

Departamento de Ingeniería Telemática

PROYECTO FIN DE CARRERA



**DESARROLLO DE UN PLUGIN DE
CHROME PARA EL ANÁLISIS DE LA
PUBLICIDAD ONLINE BASADA EN
COMPORTAMIENTO**

Autor: **Juan José Castaño Bejarano**

Tutor: **Ruben Cuevas Rumin**

Leganés, Septiembre 2015

**Título: Desarrollo de un plugin de Chrome para el análisis de la publicidad online
basado en comportamiento.**

Autor: Juan Jose Castaño Bejarano

Director: Ruben Cuevas Rumin

EL TRIBUNAL

Presidente: José Alberto Hernández Gutierrez

Vocal: Matilde Sánchez Fernández

Secretario: Javier Suárez Granizo

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 26 de Octubre de 2015 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

Matilde Sánchez Fernández
VOCAL

Javier Suárez Granizo
SECRETARIO

José Alberto Hernández Gutierrez
PRESIDENTE

Agradecimientos

Quisiera agradecer a mi mujer Verónica
su infinita comprensión

y a mi hijo Nicolás
su resignado conformismo

por todas esas horas
que este proyecto nos ha robado

Resumen

Hoy en día, quien no está en una red social, no existe.

Es una sentencia dura, pero en muchos aspectos, muy real pues según el estudio de la Online Business School de Social Media 2015, España contaba en 2014 con una población online de 23 millones de los cuales 17 millones decía utilizar activamente las redes sociales.

Desde el año 2000 hasta el 2014 la audiencia online ha crecido un 741% y empresas de todo tipo han surgido alrededor de este nuevo boom. Si bien muchas de ellas recaban información sobre el usuario medio y tratan de elaborar perfiles para ofrecer campañas de marketing o un contenido mucho más afín al usuario, se debe plantear hasta dónde es lícito permitir a dichas empresas indagar sobre una persona.

En base a los datos recabados por el tipo de contenido visitado, su asiduidad o la localización desde donde se realiza, es fácil determinar las aficiones o preferencias, pero también su condición sexual, sus creencias religiosas, afinidades políticas o si cuenta o no con alguna enfermedad.

Para intentar desentrañar cómo las empresas de publicidad online son capaces de obtener esa información y tener datos de cómo a un determinado perfil de usuario se le muestran determinadas campañas publicitarias surge la necesidad de crear un sistema que permita demostrar esta correlación de una forma empírica y denunciar este tipo de abusos.

Ahí es donde cobra sentido este proyecto. Una pieza pequeña dentro de un sistema más amplio que permita recabar información sobre la actividad de un usuario en la web y de las campañas publicitarias que le son mostradas.

Palabras clave:

privacidad, internet, cookies, tracking, marketing, plugin

Abstract

Nowadays, those who are not in a social network, do not exist.

It's a tough quote, but in many ways, very real. A study of Online Business School for Social Media 2015, Spain had in 2014 an online population of 23 million from whom 17 million said actively use social networks.

From 2000 until 2014 the online audience has grown 741% and businesses of all kinds have sprung up around this new boom. While many of them gather information about users and try to create profiles to provide marketing campaigns or more related content to that user, it should be considered where it is permissible to allow these companies to inquire about a person.

Based on data collected by the type of content visited, their attendance or the location from where it is performed, it is easy to determine the tastes and preferences, but also their sexual orientation, religious beliefs, political affiliation or whether or not it has any disease.

To try to unravel how online advertising companies are able to obtain that information and why certain advertising campaigns are shown instead of other campaigns there is a need to create a system which can demonstrate this correlation in an empirical way and denounce such abuses.

That's where this project makes sense. A small piece of a larger system that allows collecting information about user activity on the web and advertising campaigns that are displayed.

Keywords:

privacy, internet, cookies, tracking, marketing, plugin

1. Introducción y objetivos.....	1
1.1. Introducción.....	1
1.2. Objetivos	2
1.3. Metodología de trabajo	3
1.4. Medios empleados	4
1.5. Estructura de la memoria	4
2. Estado del Arte	6
3. Historias de Usuario.....	9
3.1. toma de requisitos inicial y análisis	9
3.2. Iteración 1.....	10
3.3. Iteración 2.....	10
3.4. Iteración 3.....	11
3.5. Iteración 4.....	13
3.6. Entrega Final.....	13
4. Módulos.....	14
4.1. Plugin para Google Chrome.....	15
4.1.1. Módulo "Browser"	16
4.1.2. Módulo "Activities"	18
4.1.3. Módulo "Requests"	21
4.1.4. Módulo "Trackers"	23
4.1.5. Módulo "Location"	23
4.2. Back End.....	24
4.2.1. API REST	26
4.2.2. Base de datos	28
4.3. Front End	31
4.3.1. Index	31
4.3.2. Análisis (mock up)	32
5. Planificación y Presupuesto	33
5.1. Fases del Proyecto.....	33
5.2. Personal	34
5.3. Estimación de tiempos y costes	35
6. Conclusiones y futuros desarrollos.....	36
6.1. Conclusiones	36
6.2. Futuros desarrollos.....	37
6.3. Dificultades encontradas.....	39
7. Referencias.....	40
7.1. Diccionario de Términos	40
7.2. Bibliografía.....	43

7.3. Sites.....	44
7.4. Artículos.....	44
ANEXO A - Manual de instalación de la extensión	46
ANEXO B - Manual de instalación de la aplicación servidora	51
ANEXO C - Manual de instalación del interfaz web.....	63
ANEXO D - Manual de usuario del interfaz web	65

1. INTRODUCCIÓN Y OBJETIVOS

A continuación se incluye una breve introducción al proyecto que será desarrollada más extensamente en el punto "2. Estado del arte".

Asimismo, se enumerarán los objetivos del proyecto y el proceso seguido para su consecución, profundizando en la metodología de trabajo y en los medios con los que se ha contado para llevarlo a cabo.

Para finalizar, en el último punto se recoge una sinopsis de la memoria y de la estructura de la misma.

1.1. INTRODUCCIÓN

Actualmente, cualquier usuario de internet recorre la red pasando por docenas, sino cientos, de páginas a diario donde, de una forma casi mágica, se muestran anuncios cada vez más relacionados con su personalidad, sus gustos y aficiones.

Si bien esto podría parecer ideal, pues a cada usuario se le ofrece un contenido cada vez más customizado, este derecho a tener contenidos cada vez más afines a nuestros gustos choca frontalmente con el derecho a la privacidad que cada persona tiene.

Y es que, si bien cuando una persona se registra en una página y ésta le solicita sus datos personales... ¿alguien se pregunta para que los quieren? ¿Por qué al darse de alta en un foro (por ejemplo, sobre cine) se solicita la edad, el estado civil o el número de teléfono? ¿Es esto realmente relevante o hay alguna finalidad oculta detrás de ello?

A parte de los datos que cualquier simple registro pueda solicitar, es mucho más preocupante si cabe lo que el usuario revela de forma más o menos voluntaria. Sus gustos, inquietudes, preferencias,.. sus fotos, videos,.. en definitiva, su vida al completo.

Por último, otro tipo de información susceptible de dejar rastro es aquella que se genera simplemente al navegar de un sitio a otro. Qué páginas se visitan, qué contenido se consulta, cuánto tiempo se dedica a ello e incluso desde donde.

1.2. OBJETIVOS

La finalidad de este proyecto es poner la primera piedra de un montaje mucho más grande que permitirá contrastar de una forma empírica si la publicidad expuesta a determinado perfil de usuario viene dirigida en base a las páginas que ha visitado, su gustos y aficiones, con lo que las campañas de publicidad estarían vulnerando el derecho a la privacidad de cada individuo.

En lo referente a este proyecto, los objetivos estaban marcados desde un inicio, aunque han sido modificados sobre la marcha, según ha ido avanzado el desarrollo del mismo.

El objetivo inicial del proyecto era la creación de un plugin sobre algún navegador con alta penetración en el mercado y que permitiese monitorizar las peticiones que se realizan durante una navegación estándar.

Dicha información será valiosa para extraer de ella los saltos que se producen en los anuncios presentes en la mayoría de las páginas web y conocer de primera mano tanto los mediadores que existen en las campañas publicitarias como el site final en el que reside el anuncio presentado.

Con el avance del proyecto, se han incluido diversos puntos que son listados a continuación:

- El plugin será realizado sobre Google Chrome
- Se realizará un servicio que permitirá obtener la IP del usuario y que, a través de un servicio de geolocalización, obtendrá la localización física de la máquina que se está monitorizando.
- Se incluirá un servicio que permitirá monitorizar la asignación de cookies y que guardará dicha información junto a la información anterior
- Se añadirá un servicio que guardará el histórico del navegador hasta el momento de inicialización del plugin, consolidando dicha información de forma distinta para poder saber qué páginas fueron visitadas previamente por el usuario y cuales se visitaron con monitorización activa del plugin.
- Se desarrollará la parte servidora, registrando ésta la información recolectada por el plugin

- Se persistirá dicha información en una base de datos relacional
- Se deberán poder visualizar los datos recolectados por el plugin a través de una aplicación web sencilla

Estos requisitos se exponen de una forma más amplia en el punto "3. Historias de Usuario". En él se desglosan en formato "historias de usuario" para poder acometer su desarrollo en diferentes iteraciones y poder enseñar y contrastar la funcionalidad de una manera ágil con el stakeholder.

1.3. METODOLOGÍA DE TRABAJO

Dado que el alcance inicial del proyecto no estaba definido al inicio del mismo, se ha optado por utilizar una metodología ágil para su desarrollo.

Las metodologías ágiles (conocidas también como "agile") están basadas en el desarrollo iterativo e incremental, donde los requisitos van cambiando durante el desarrollo del mismo dependiendo de las necesidades del cliente. De este modo, éste obtiene cada cierto tiempo una pieza pequeña pero funcional que le ayuda a ver si realmente lo que se planteó como un requisito en la anterior iteración es lo que necesita y, si es así, centrar el foco para la siguiente lista de requisitos.

En el punto "7. Referencias" se incluye varios libros de lectura muy recomendable al respecto.

Dado que en dicho proyecto sólo ha participado un stakeholder y un desarrollador, no se podría englobar la metodología seguida en ninguna de las clásicas (scrum, kanban,..) aunque sí que se ha seguido el manifiesto ágil:

Individuos e interacciones sobre procesos y herramientas

Software funcionando sobre documentación extensiva

Colaboración con el cliente sobre negociación contractual

Respuesta ante el cambio sobre seguir un plan

1.4. MEDIOS EMPLEADOS

Dado que los objetivos del proyecto se restringen a la creación de un plugin y al envío de datos a un servidor para poder mostrarlos, el montaje (que podría haber sido todo lo complejo que se requiriese) se ha simplificado para poder levantar todos los módulos en la misma máquina y poder utilizar un único ordenador para todo.

En este caso se ha utilizado un Macbook Pro 13" con 4GB de RAM y un procesador i7 de finales de 2011.

A efectos de cálculos de gastos para el proyecto, se ha contado además con una conexión de fibra óptica a internet de 50 Mb y una línea telefónica.

Según se vayan añadiendo nuevos módulos para componer la aplicación final, el montaje deberá ser más robusto y requerirá de un servidor propio o el montaje del mismo en algún PaaS (Heroku, Open Shift,...).

En ambos casos, dependiendo del tamaño de usuarios la parte servidora deberá tener la posibilidad de escalar, bien montando clusters (Si se ha optado por montar un servidor físico) bien aumentando el número de gears/máquinas virtuales del PaaS elegido.

1.5. ESTRUCTURA DE LA MEMORIA

La memoria se compone de:

- Un prólogo/resumen en dos idiomas (Castellano e Inglés) donde se hace una descripción general del proyecto
- Una introducción donde se exponen los objetivos del proyecto y un breve comentario sobre cómo ha sido desarrollado
- Una exposición del estado del arte actual donde se trata de arrojar luz sobre la publicidad en internet y cómo ésta vulnera (presuntamente) los derechos de privacidad individuales

- Una explicación exhaustiva sobre los distintos módulos desarrollados, siendo éstos el plugin de Chrome, la parte servidora que guarda los datos y la parte front que los muestra
- Un presupuesto económico para el desarrollo del proyecto
- Las conclusiones y los futuros desarrollos que se deben realizar
- Una referencia de términos y otra bibliográfica, donde poder consultar tanto las palabras utilizadas en la memoria como los conceptos descritos en la misma
- Un conjunto de 3 anexos en los que se describe la instalación del plugin, la parte servidora y un manual de usuario de la aplicación front

2. ESTADO DEL ARTE

El derecho a la privacidad es un derecho fundamental del que, hoy en día, es difícil disfrutar.

En un mundo conectado, donde si no estás en una o varias redes sociales, no existes y donde si no consumes contenido online no estás a la última, la posibilidad de no estar expuesto a algún ataque se reduce prácticamente a cero.

La actividad de determinada persona en la red siempre deja un rastro que se podría englobar en uno de los siguientes grupos:

Datos aportados aportados voluntariamente

Son aquellos que se dan voluntariamente al registrarse en determinado sitio de internet como pueden ser el nombre y los apellidos, el DNI, teléfono, tarjetas de crédito..

Son importantes porque dicen quién es el usuario y permiten identificarle.

El mayor riesgo que esto implica es que, en base a estos datos, una persona podría llegar a suplantar a otra. Otros riesgos de menor consideración podrían ser que se podrían utilizar para:

- Tener una forma de contactar o localizar a la persona
- Sugerir su procedencia y sus circunstancias sociales
- Recolectar sus hábitos, aficiones y preferencias
- Conocer su entorno familiar o afectivo

A la hora de revelar estos datos es importante conocer la política de privacidad del sitio web. Legalmente están obligados a exponer dicha política públicamente y forzar al usuario a leerla y aceptarla antes de ingresar en dicho sitio, aunque normalmente es tan confusa, tediosa y con un lenguaje tan técnico que suele pasarse por alto.

Datos publicados por terceros

Son aquellos que el usuario libremente comparte a través de redes sociales, foros o portales de blogging o video-blogging pero que, al poco tiempo.

Otra tipología de datos públicos serían aquellos que se difunden a través de medios oficiales (como puedan ser a través de la administración) o de medios de información públicos (como puedan ser periódicos, revistas, radio o televisiones)

Dado que estos datos se publicitan en sitios con alta concentración de datos es un nicho donde los motores de búsqueda recaban información indexándola y difundiéndola aún más.

Legalmente asiste el derecho a que se cancelen y borren dichos datos si no cuentan con la aprobación de la persona en cuestión, quien debería dirigirse al responsable del sitio web que aloja el contenido o al motor de búsqueda que lo difunde.

En este caso, el derecho a la privacidad en ocasiones choca de frente al derecho a la información.

Datos de navegación y de comportamiento en la red

Estos datos son recabados sin que el usuario sea consciente de ello y, en muchas ocasiones, sin que se le haya informado mínimamente qué se está haciendo y, sobre todo, con qué finalidad.

Estos datos pueden ser la IP de la máquina, con la cual se podría geolocalizar desde donde se conecta el usuario o a través de qué proveedor lo realiza.

Así mismo, a través de las famosas cookies o del "tracking pixel", se pueden detectar o trackear el comportamiento de determinada persona en la web, estudiando el número de visitas que hace a determinadas páginas, con qué frecuencia,.. y con ello poder elaborar un perfil de usuario para determinar su gustos, afinidades, condición sexual, religión, enfermedades y es ahí donde se llega a la difusa línea entre qué datos son lícitos que una empresa explote a través de minería de datos para ofrecer determinada campaña de marketing y qué datos vulneran la privacidad del usuario.

El conjunto de los tres grupos mencionadas es a lo que se podría llamar el rastro o la huella digital que el usuario deja en internet.

La pregunta que debe hacerse es.. ¿Por qué hay empresas de internet que gastan millones en crear y gestionar sites que luego ofrecen de modo gratuito?

La respuesta es que, cuando un servicio es gratuito, es porque el usuario no es el cliente, si no el producto que se vende.

3. HISTORIAS DE USUARIO

Los requisitos que se tratan en metodologías ágiles se denominan historias de usuario, siendo estos una frase más o menos reducida que pretende reflejar *quién* pide *qué* y *para qué*.

La filosofía que hay detrás de una historia de usuario sigue el principio INVEST ya que una historia debe ser:

- **I**ndependent: Debe de ser independiente y tener sentido por si misma
- **N**egotiable: Se pueden negociar, priorizar e incluso cambiar
- **V**aluable: Debe de aportar un valor de negocio al usuario
- **E**stimable: Se debe poder estimar
- **S**mall: Deben de ser pequeñas para poder ser estimables e independientes
- **T**estable: Se deben de poder probar

Dada la naturaleza iterativa e incremental que las metodologías ágiles promueven, a continuación se expondrá el listado de iteraciones con el que ha contado el proyecto y las historias de usuario que se han incluido en cada una de ellas.

3.1. TOMA DE REQUISITOS INICIAL Y ANÁLISIS

Como inicio del proyecto, se mantiene una primera reunión con el stakeholder en el que se definen las líneas globales del proyecto.

Tras la reunión se realiza el análisis del proyecto que ha sido expuesto por el stakeholder y se plantea un conjunto de historias de usuario que pueden abordarse en una primera iteración.

Se intenta hacer un desarrollo rápido del plugin de chrome que permita enviar información a un servidor para plasmar cuales serán las bases de futuros desarrollos en próximas iteraciones.

3.2. ITERACIÓN 1

Como	tutor del proyecto
Quiero	un plugin para un navegador con alta penetración en el mercado que me permita monitorizar las peticiones que se realizan en un browser cuando se visita una página con anuncios
Para poder	montar un sistema posteriormente que consuma dichos datos y que me permita trazar una relación causa/efecto entre la tipología de páginas visitadas y la tipología de anuncios que son mostrados al usuario

Como resultado de esta historia de usuario se entregó un plugin de Google Chrome que permitía monitorizar las peticiones de recursos/url's que se realizaban desde una página con anuncios.

Como	tutor del proyecto
Quiero	poder compartir la información extraída por el plugin con otros sistemas
Para poder	montar el ecosistema citado en la historia anterior

Como resultado de esta historia de usuario se entregó un servidor desarrollado sobre java y ejecutable sobre un Tomcat que exponía un API REST que permitía enviar información sobre url's y recursos, manteniendo dicha información en memoria. Para poder visualizar los datos, se implementó una página que hacía el volcado de dicha información en formato tabla.

3.3. ITERACIÓN 2

Como	tutor del proyecto
Quiero	poder serializar la información enviada al servidor en una base relacional
Para poder	utilizar dicha información posteriormente desde un sistema mucho más complejo

Como resultado de esta historia de usuario se entregó una modificación sobre el servidor original que permitía serializar la información que antes se contenía en memoria en una base de datos MySQL.

Como	tutor del proyecto
Quiero	poder tener un identificador único por plugin
Para poder	discernir qué peticiones de páginas hace cada usuario y qué recursos/url's/anuncios se le muestra a cada uno

Como resultado de esta historia de usuario se generó un módulo en el plugin que permitía generar un identificador único y que era enviado al servidor junto a la información de navegación

Como	tutor del proyecto
Quiero	poder saber cuando un plugin es desinstalado y poder notificarselo al servidor
Para poder	conocer cuantos plugins hay activos en cada momento

Tras investigar en diversas fuentes la posibilidad de suscripción al momento de desinstalación de un plugin se vio que técnicamente no era posible acometer dicha historia de usuario, con lo que no fue realizada.

3.4. ITERACIÓN 3

Como	tutor del proyecto
Quiero	poder relacionar un browser con la url de una página y la actividad que se produce dentro de ella
Para poder	determinar por un lado que visitas realiza el usuario (que determinarán su perfil e intereses) y, por otro lado, cuales son los anuncios que se le muestran

Como resultado de esta historia de usuario se dividió el modelo de la base de datos en tres tablas. Una tabla para los browsers con su identificador y el timestamp de creación y de última actualización. Otra tabla con las url's que dicho browser ha visitado. Otra tabla con la información de todos los recursos que se han descargado para dicha página.

Dado que el ámbito de dicho proyecto no recogía el conectarse con el servicio de resolución de rutas para identificar si una url presente en la página era de un anuncio o no, se optó por enviar todas las peticiones monitorizadas para que posteriormente fuesen filtradas cuando se desarrollase esa parte.

Como	tutor del proyecto
Quiero	poder saber qué entidades están poniendo cookies en la páginas
Para poder	identificar posteriormente si dichas entidades son responsables de vulnerar la privacidad del usuario de dicha página

Como resultado de esta historia de usuario se añadió un nuevo módulo al plugin de Chrome y su correspondiente representación en el servidor para permitir monitorizar la creación/modificación de cookies y que dicha información fuese enviada junto a los datos de navegación para que persistiesen también en la base de datos.

Como	tutor del proyecto
Quiero	poder saber desde donde se realizó la petición
Para poder	discriminar si la localización del usuario también es un criterio que los motores de anuncios tienen en cuenta para lanzar las distintas campañas

Como resultado de esta historia de usuario se investigó inicialmente si se podría solicitar la IP de la maquina al API de Extensiones de Chrome. El API no lo permitía.

Así, se añadió un nuevo módulo al plugin de Chrome que permitiese geolocalizar el navegador realizando una llamada REST a un servicio online de geolocalización de modo que, cuando se enviase la información al servidor, se enviase también desde qué sitio se realizó dicha navegación.

Como	desarrollador de la solución
Quiero	poder enseñar una posible visualización de la información recolectada
Para poder	realizar una presentación entendible, aunque aún la solución no estuviese enganchada con el sistema final.

Como resultado de esta historia de usuario, se realizó una página que permitía seleccionar un rango de fechas para las peticiones de páginas que había realizado el usuario y un rango de fechas para los anuncios que habían sido mostrados. En base a ese filtro, se visualizaría las distintas categorías en un gráfico para poder analizar si ambos datos tenían correlación o no. Dado que es un mockup, los datos están hard coded en el servidor.

3.5. ITERACIÓN 4

Como	tutor del proyecto
Quiero	tener el código documentado
Para poder	continuar con el desarrollo iniciado con otra persona

Como resultado de esta historia de usuario se ha revisado el código y se ha ido documentando línea a línea todo el código java y js.

Como	tribunal del proyecto
Quiero	tener una memoria que explicite los distintos módulos de los que consta el proyecto, así como información relativa al porqué del mismo
Para poder	evaluar el trabajo desarrollado

Como resultado de esta historia de usuario se ha realizado esta memoria, en la que se recoge tanto la información del porqué del proyecto como la información de cómo se ha realizado.

3.6. ENTREGA FINAL

Como	tribunal del proyecto
Quiero	que el alumno realice una defensa del proyecto
Para poder	evaluar el trabajo desarrollado

Como resultado de esta historia de usuario se realizará una demostración del funcionamiento del plugin, así como de la parte servidora.

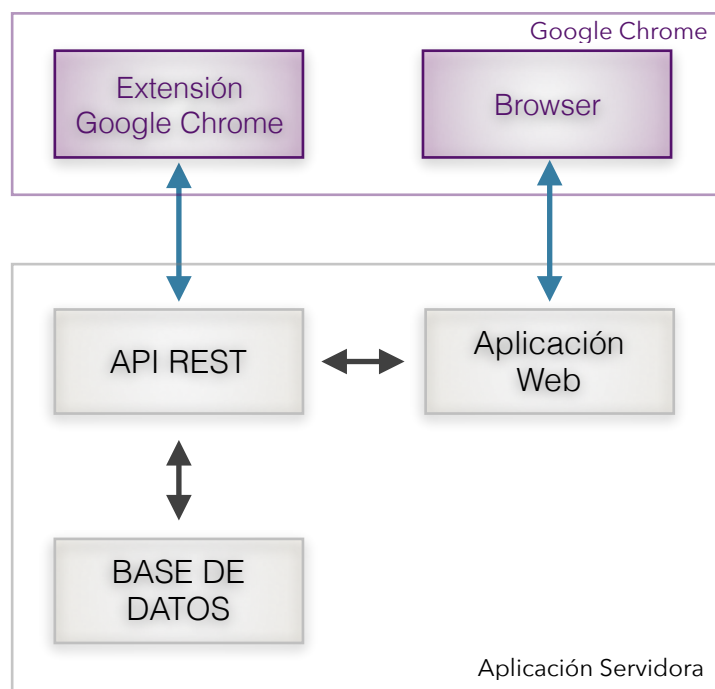
4. MÓDULOS

La aplicación se compone de varios módulos bien diferenciados.

Un plugin de extensión para Google Chrome que envía los datos al servidor utilizando un API REST.

La aplicación servidora serializa los recursos creados/actualizados mediante REST en una base de datos.

Por otro lado, para poder consultar la información que se ha recolectado, la aplicación web se sirve del mismo API REST para buscar la información y mostrársela al usuario.

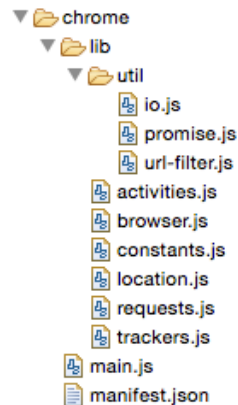


Para la correcta comprensión de los puntos siguientes, se recomienda leer con detalle la descripción de los términos: Browser, Activity, Request y Tracker que se recogen en el punto "7.1 Diccionario de términos".

4.1. PLUGIN PARA GOOGLE CHROME

Se ha desarrollado un plugin para Google Chrome que se ejecuta en background y que, por lo tanto, no interfiere con la navegación del usuario.

La estructura de dicho plugin es la siguiente:



El plugin carga una serie de scripts localizados en la carpeta lib tal y cómo se puede ver en el descriptor del plugin (*manifest.json*)

```
{
  "name": "Privacy Plugin",
  "version": "1.0",
  "manifest_version": 2,
  "description": "Este plugin recolectará la información de navegación del browser y lo enviará a un servidor para su posterior tratamiento",
  "background": {
    "scripts": [
      "lib/constants.js",
      "lib/util/io.js",
      "lib/util/promise.js",
      "lib/util/url-filter.js",
      "lib/location.js",
      "lib/activities.js",
      "lib/requests.js",
      "lib/trackers.js",
      "lib/browser.js",
      "main.js"
    ]
  },
  "permissions": [
    "webNavigation",
    "notifications",
    "pageCapture",
    "activeTab",
    "alarms",
    "sessions",
    "tabs",
  ]
}
```

```

    "history",
    "storage",
    "webRequest",
    "cookies",
    "http://*/",
    "https://*/"
  ]
}

```

La carga de dichos scripts se limita a inicializar una serie de módulos en memoria, aunque no se realiza ninguna acción hasta la ejecución del script *main.js*, quien se encarga de inicializar los distintos módulos del plugin de la siguiente forma:

```

(function () {
  "use strict";
  // Inicializamos el plugin, para ello, buscamos el id único con initUID
  BROWSER.initUID().then( function( uid ){
    // Una vez obtenido el uid, inicializamos el history, ads y location
    ACTIVITIES.startup();
    REQUESTS.startup();
    LOCATION.startup();
    TRACKERS.startup();
  });
})();

```

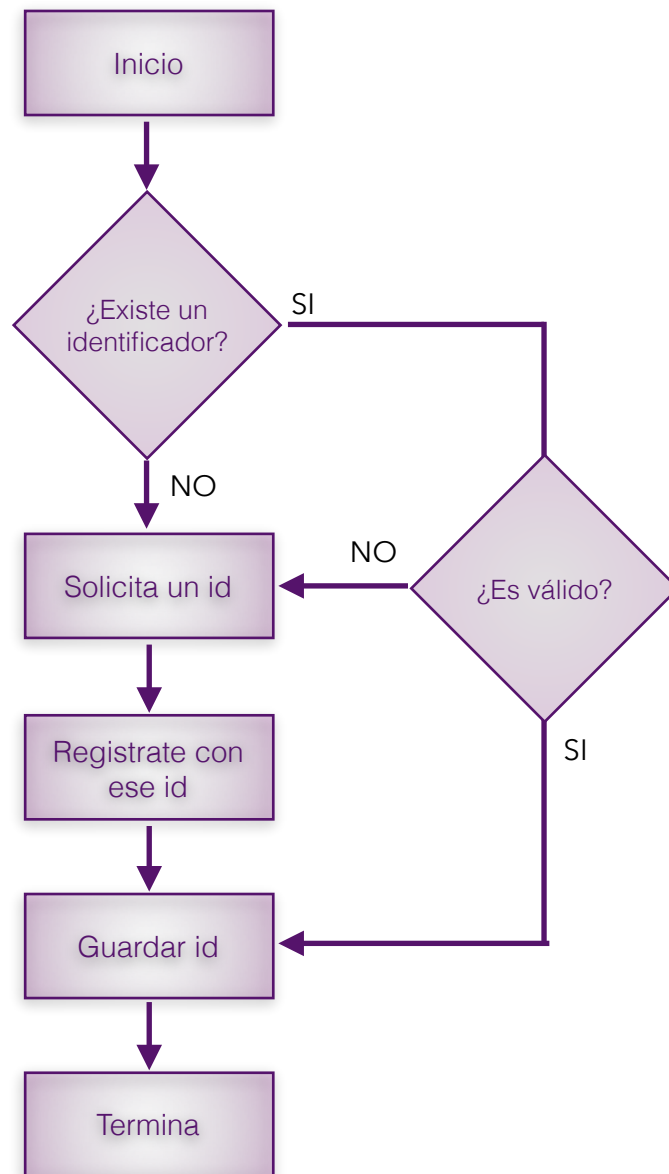
En él podemos observar cómo se obtiene un identificador único para este plugin (la primera vez lo solicitará a la parte servidora y se lo guardará en el "local storage" para poder trazar las distintas sesiones del mismo usuario en la parte servidora)

Una vez que ya se ha inicializado el módulo del browser y se tiene un identificador único, se procede a levantar el resto de módulos cuya funcionalidad estará descrita a continuación:

4.1.1. Módulo "Browser"

El módulo Browser es la parte principal del plugin y es quien se encarga de gestionar que el id del plugin sea único con respecto al resto de plugins instalados en otras máquinas.

El flujo es el siguiente:



De forma asíncrona, el plugin intenta obtener el ID desde el "local storage". Si existiese alguno guardado, se realiza una comprobación contra el servidor para ver si éste es válido.

En el caso en el que no existiese en el local storage (porque es la primera vez que se instala) o en el caso en el que no fuese válido (porque el servidor no tuviese registros previos de dicho plugin), se solicita un nuevo identificador a la parte servidora y, una vez obtenido, se le indica a ésta que se quiere crear una nueva sesión con dicho identificador.

Una vez que el plugin tiene un identificador válido, bien porque el servidor generó uno nuevo o porque el plugin pudo obtenerlo correctamente desde el "local storage" el plugin guarda dicho id en memoria y permite continuar la ejecución resolviendo la promesa y permitiendo la ejecución del resto de módulos

4.1.2. Módulo "Activities"

El módulo de actividad es el encargado de gestionar las peticiones de páginas en el browser. Tratando tanto las Requests (Peticiones de recursos/urls que se realizará en la página) como los trackers (*cookies*). Cuenta con varios sub-procesos que son lanzados desde la función startup.

El proceso principal es el que se encarga de enviar las Requests y los Trackers al servidor:

- Actualización del Servidor

Un proceso secundario es el encargado de revisar el histórico del browser para incluir también esta información en el servidor.

- Actividad Previa (historial del browser)

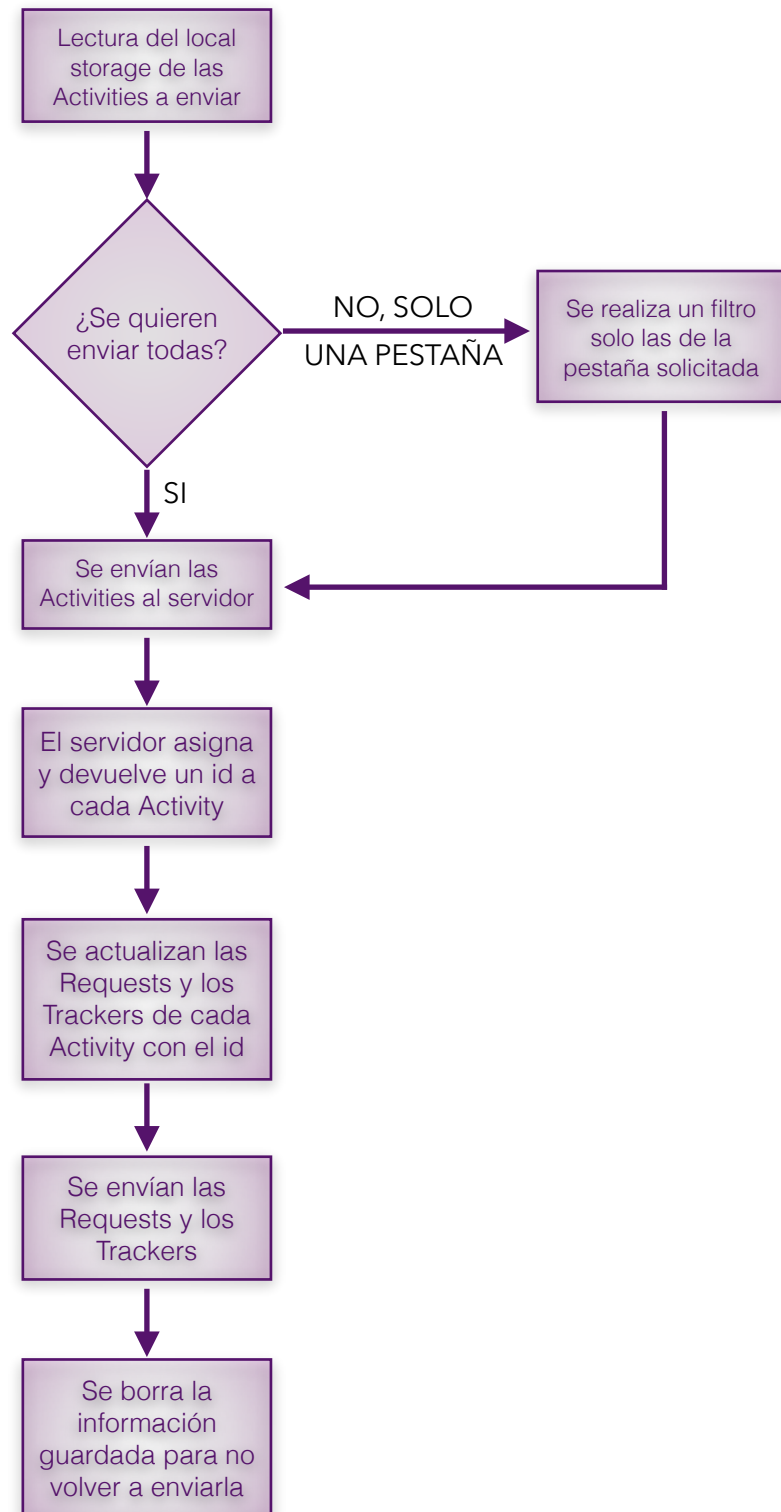
El último proceso es el encargado de la gestión de cuándo se realiza dicho envío, dependiendo de la actividad existente en el browser en ese momento.

- Monitorización de Pestañas/Browser

Actualización del Servidor

Este sub-proceso termina siendo llamado desde el sub-proceso de "actividad local previa" y desde el de "monitorización de pestañas". Se sirve de la información persistida en el local storage del browser por el módulo de Requests y el de Trackers para enviar la información al servidor.

El flujo que sigue es el siguiente:

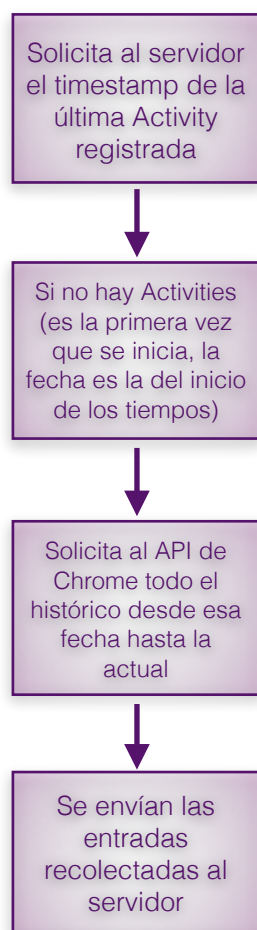


Al iniciar el módulo Activities, éste detecta si en la anterior sesión se dejaron cosas por enviar (i.e.: se cerró abruptamente la sesión). Si es así, envía toda la información guardada en el local storage al servidor. Si no, se irá enviando pestaña por pestaña.

Actividad Previa (Historial del browser)

Se encarga de recolectar la información existente en el historial del browser y revisar si debe o no enviar dicha información al servidor.

El flujo que sigue es el siguiente:



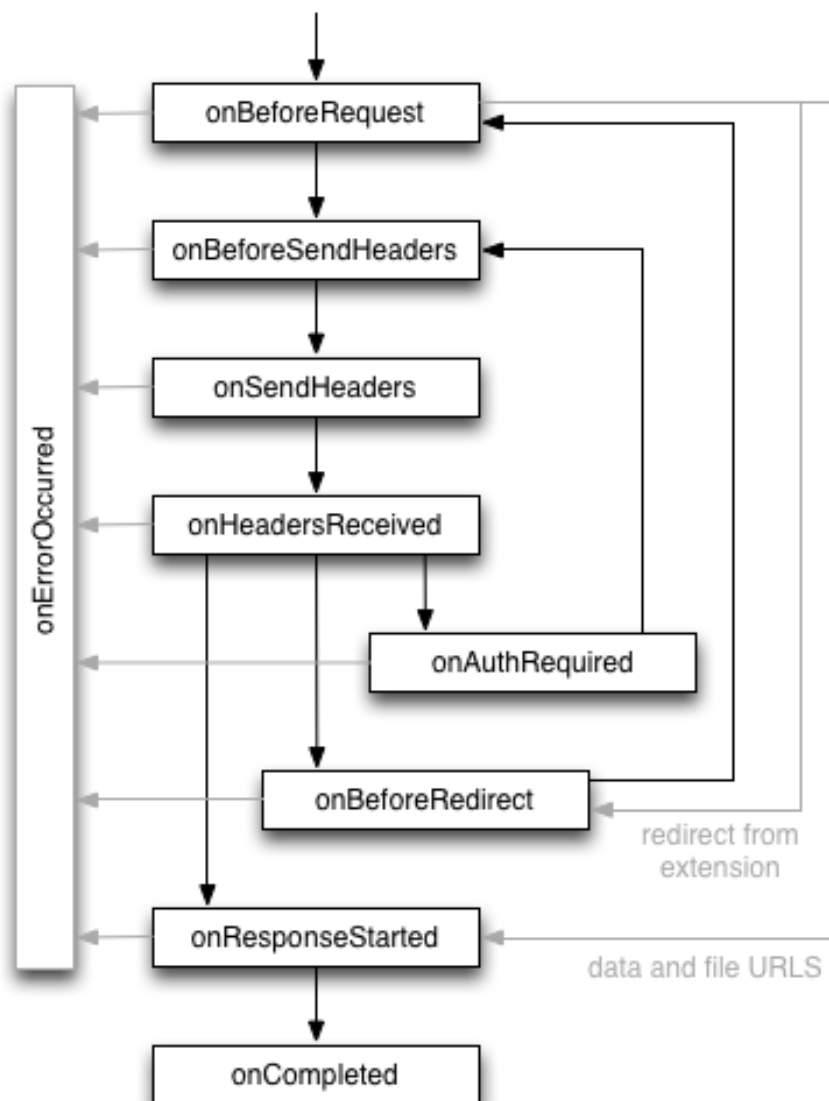
Módulo de Monitorización de Pestañas/Browser

Se encarga de monitorizar tanto las pestañas del browser como el browser en sí y, cuando se cierra (una pestaña o el browser en sí mismo) envía la información de navegación que tiene guardada de la navegación que se acaba de producir

4.1.3. Módulo "Requests"

El módulo de "Requests" es el encargado de monitorizar las peticiones de recursos y/o urls que se producen en el browser.

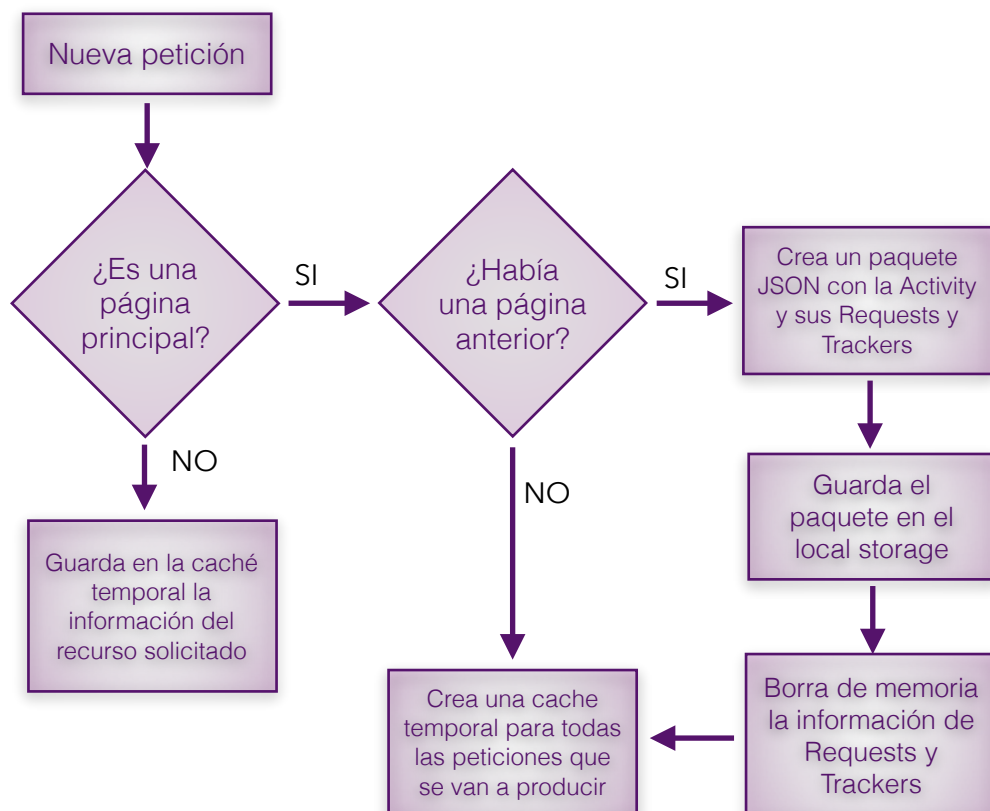
Para ello se sirve del servicio *webRequest* del API de extensiones de Google Chrome que permite monitorizar los siguientes estados de una petición:



La monitorización se realiza para todos los eventos de la figura anterior aunque, funcionalmente, se diferencian tan sólo tres, siendo estos:

- Inicialización de una petición de página: se inicializa un nuevo modelo para la petición en curso y se revisa si la pestaña en la que se está iniciando la nueva petición tenía ya un modelo activo. Si es así, lo guarda en el local storage, listo para ser enviado al servidor.
- Progreso de una petición de un recurso/url interno a una página. Se va guardando en el modelo de la página principal el progreso de dicha petición para poder listar todos los recursos descargados y el proceso que se ha seguido para poder solicitarlo (saltos, IP's,...)
- Finalización de una petición, bien porque se haya producido un error o porque el elemento haya terminado de cargarse.

El flujo que sigue es el siguiente:



4.1.4. Módulo "Trackers"

El módulo de trackers se sirve del servicio de monitorización de cookies que provee el API de extensiones de Google Chrome y que permite conocer cuándo se están creando/modificando las cookies presentes en la página.

Esta información será relevante para conocer qué páginas utilizan cookies y que uso se pueden estar haciendo de las mismas por las distintas campañas de publicidad.

Modificación de cookies

Cuando se están modificando cookies en la página, el API de chrome ofrece la posibilidad de registrarse a este hecho y poder recolectar la información sobre qué dominio está registrando una determinada cookie, así como su nombre y valor.

Este subproceso recolecta esa información y la guarda en una caché interna indexada con el identificador de la pestaña donde se ha puesto dicha cookie. Por otro lado, provee de métodos de acceso y borrado de dicha caché para que, cuando el módulo de recolección de información de Requests esté montando la información a enviar al servidor, pueda adjuntar a una Activity tanto las Requests como los Trackers.

4.1.5. Módulo "Location"

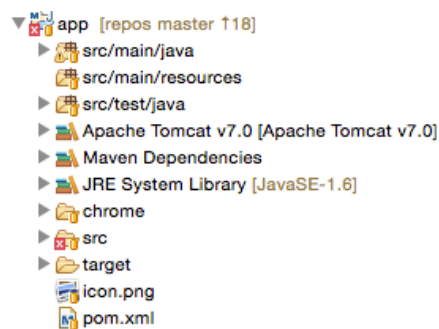
El módulo de localización es el más pequeño de todos y es el encargado de activar una alarma que, cada cierto periodo de tiempo, se encarga de realizar una petición a un servicio online parametrizado por configuración que permite geolocalizar el browser y devuelve información relevante como pueda ser su IP, su longitud, latitud...

La información de la última localización resuelta la deja en una variable del modulo que podrá ser accedido desde el modulo de Request para que, cuando éste esté montando la información a enviar al servidor, pueda adjuntar la información de la IP, la longitud y la latitud a la Activity para poder conocer desde donde se realizó dicha petición.

4.2. BACK END

Se ha implementado una aplicación en el lado del servidor que cuenta con un módulo de API REST que permite crear/modificar los distintos recursos. Se entienden cómo recursos las distintas entidades que forman parte de un modelo que será serializado/deserializado en una base de datos MySQL

El proyecto Java en el que está englobado el módulo de back end se ha desarrollado sobre Eclipse y tiene la siguiente estructura:

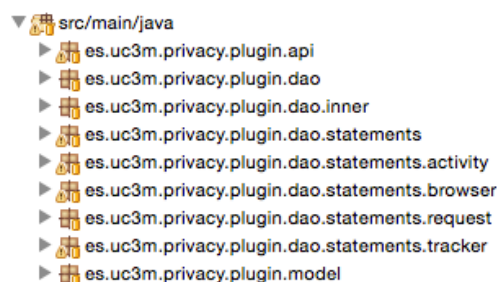


En ella cabe destacar que la estructura es la propuesta por maven (en la raíz existe un pom.xml que será quien marque las dependencias del proyecto y cómo se construye éste)

Por otro lado, existen las carpetas *src/main/java* que es donde se contienen los fuentes de la aplicación y las carpetas *src/main/resources* donde se incluirían ficheros de configuración (en este caso, ahí estará el config.properties con la configuración de la BBDD)

Por último, en la carpeta *src/test/java* se podrían crear tests automáticos que se ejecutarían antes de realizar la build del war. En este caso, no se han requisitado tales tests

Profundizando en la estructura del código fuente, ésta presenta los siguientes paquetes:



es.uc3m.privacy.plugin.api

Contiene los recursos REST que conforman el API de intercambio

es.uc3m.privacy.plugin.dao

Contiene una clase DAO que es quien concentra el intercambio de datos con la base de datos subyacente (MySQL en este caso) devolviendo un interfaz a modo de singleton del tipo `IContentProvider`, que describe las posibles acciones sobre el modelo de la base de datos (búsquedas, inserciones,...)

Se ha optado por la creación de este DAO para dotar al sistema de una primera capa de abstracción que permitiese devolver una implementación u otra de `IContentProvider`, dependiendo del tipo de base de datos con el que se contase finalmente.

A modo de reseña, en fases iniciales del desarrollo, cuando aún no estaba claro qué base de datos usar y todo se guardaba en memoria, este DAO devolvía una clase dummy que guardaba y devolvía toda la información que estaba registrada en objetos `HashMap`.

es.uc3m.privacy.plugin.dao.statements

Contiene la implementación de `IContentProvider` y una factoría que devolverá, dada una constante, la sentencia que se quiere ejecutar (inserción, actualización, creación) sobre un elemento dado, como si de un comando se tratase, en el que se indica qué acción se quiere ejecutar y con qué parámetros.

Se ha optado por este modelo para poner dotar al sistema de una segunda capa de abstracción sobre la anterior. De este modo, se podrían desarrollar distintas queries sobre un mismo comando (Por ejemplo para hacerlas más eficientes, o tener queries dummies para poder realizar pruebas automáticas)

La implementación de todas las sentencias SQL están recogidas dentro de cada uno de sus subpaquetes, dependiendo de la tabla que se ve afectada.

es.uc3m.privacy.plugin.dao.model

En este paquete se contienen los modelos de cada entidad y que serán los intercambiados por el API REST.

4.2.1. Api Rest

El API REST se ha desarrollado en Java sobre un Tomcat 7, utilizando la librería Open Source Jersey. Se ha optado por su utilización por la sencillez de la misma y la buena documentación existente.

El API REST se divide en los siguientes recursos:

BrowserResource

Se trata del recurso principal que modela cada una de las instancias del plugin y permite las siguientes operaciones:

- **newBrowser:** Crea un nuevo browser con el identificador {brwId}
POST --> <http://server:port/api/browser>
- **getBrowser:** Devuelve el browser con el identificador {brwId}
GET --> <http://server:port/api/browser/{brwId}>
- **getBrowsers:** Devuelve el listado de browsers
GET --> <http://server:port/api/browser>
- **getBrowserIds:** Devuelve el listado de browsers ids
GET --> <http://server:port/api/browser>
- **getNewId:** Crea un nuevo identificador único de browser
GET --> <http://server:port/api/browser/newId>
- **getCount:** Devuelve el número de browsers registrados
GET --> <http://server:port/api/browser/count>
- **getBrowserActivities:** Devuelve las actividades del browser {brwId}
GET --> <http://server:port/api/browser/{brwId}/activity>
- **getActivitiesBetween:** Listado de todas las actividades de un browser en un rango de fechas concreta
GET --> <http://server:port/api/browser/{brwId}/activities/between/{dateStart}/and/{dateEnd}>

- **getActivitiesLiveBetween:** Listado de las actividades guardadas con el plugin en acción para un browser y en un rango de fechas concreto

GET --> <http://server:port/api/browser/{brwId}/live-activities/between/{dateStart}/and/{dateEnd}>

ActivityResource

Se trata del recurso que modela cada una de las peticiones que realiza el browser. Permite las siguientes operaciones:

- **insertActivities:** Inserta en la base de datos las actividades pasadas por parámetro

PUT -> <http://server:port/api/activity>

- **getActivityForBrowser:** Devuelve el listado de actividad para el browser {brwId}

GET --> <http://server:port/api/activity/browser/{id}>

- **getMostRecentEntry:** Devuelve la ultima actividad del browser con el identificador que se ha pasado por parámetro {brwId}

GET --> <http://server:port/api/activity/browser/{id}/last-activity>

- **getRequestsForActivity:** Devuelve el listado de Requests de la actividad con el id del parámetro

GET --> <http://server:port/api/activity/{activityId}/requests>

- **getTrackersForActivity:** Devuelve el listado de Trackers de la actividad con el id del parámetro

GET --> <http://server:port/api/activity/{activityId}/trackers>

RequestResource

Se trata del recurso que modela cada una de los recursos/url's que se descargan durante una navegación a una página determinada. Permite las siguientes operaciones:

- **insertRequests:** Inserta el listado de Requests en la base de datos

PUT --> <http://server:port/api/request>

TrackingResource

Se trata del recurso que modela cada uno de los trackers y permite las siguientes operaciones:

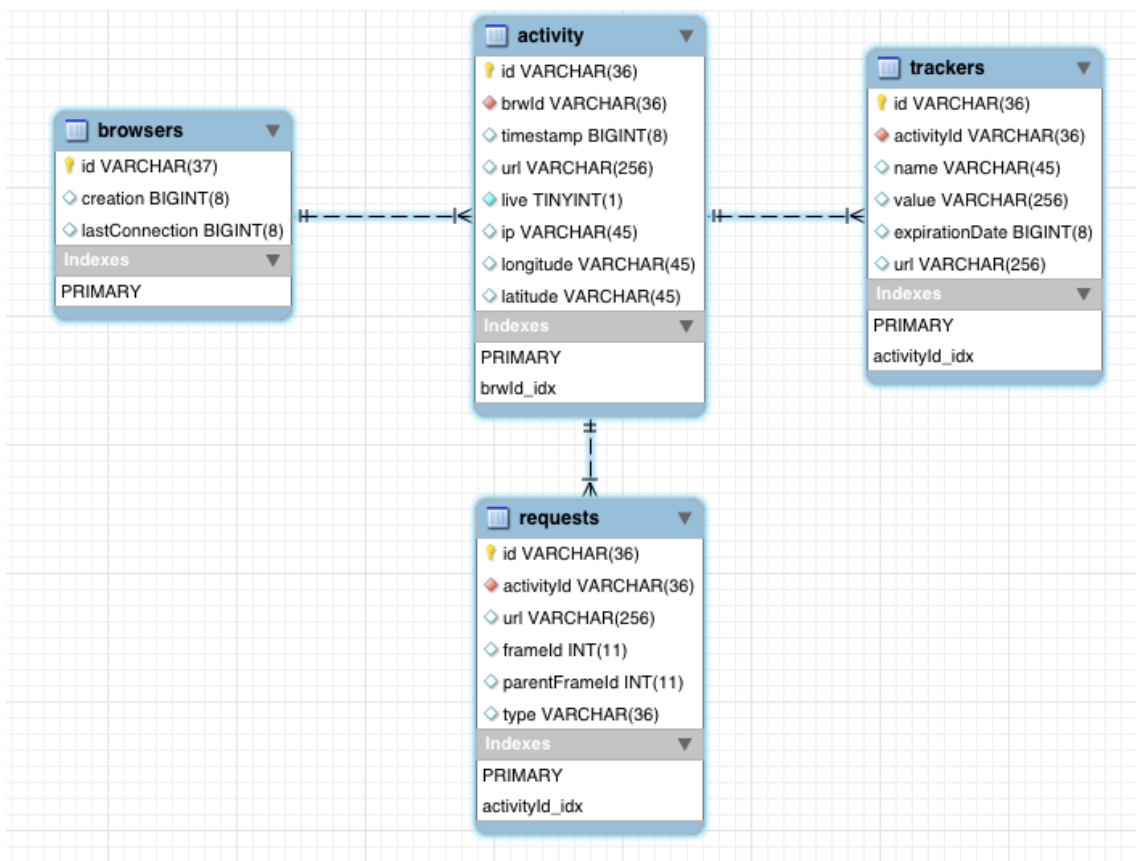
- **insertRequests:** Inserta el listado de trackers en la base de datos

PUT --> <http://server:port/api/tracker>

4.2.2. Base De Datos

Dado que no había requisitos al respecto, se ha elegido como base de datos una MySQL pues es una de las base de datos más extendidas para desarrollos de estas características. Dependiendo de la concurrencia de peticiones y la carga de la misma puede ser recomendable pensar en alguna otra base de datos más robusta.

El modelo de tablas que se ha creado va alineado con el de los recursos REST anteriormente expuestos:



En él se puede observar como se almacenan todos los datos relevantes de cada entidad y cómo se relacionan unos con otros para poder realizar consultas que permitan obtener el resultado requerido en cada ocasión.

browsers

- **id**: Identificador único de 32 caracteres
- **creation**: Timestamp de tipo long con la fecha de dada de alta en el sistema
- **lastConnection**: timestamp de tipo long con la fecha de último acceso al sistema

activity

- **id**: Identificador único de 32 caracteres
- **brwld**: Identificador único del browser de 32 caracteres
- **timestamp**: Timestamp de tipo long con la fecha de petición de dicha url
- **url**: URL que se ha solicitado en el navegador (Si es superior a 255, se trunca)
- **live**: Booleano que indica si la petición se hizo con el plugin activo o si proviene del histórico del browser
- **ip**: IP desde donde se realizó la petición
- **longitude**: Geolocalización desde donde se hizo la petición
- **latitude**: Geolocalización desde donde se hizo la petición

requests

- **id**: Identificador único de 32 caracteres
- **activityId**: Identificador único de actividad de 32 caracteres
- **url**: URL del recurso/dirección que se está solicitando (Si es superior a 255, se trunca)
- **frameId**: Identificador del frame que realiza la petición (-1 si es el frame principal)
- **parentFrameId**: Identificador del frame padre (-1 si es el frame principal)
- **type**: Tipo de recurso que se solicita (script, frame, image,...)

trackers

- **id**: Identificador único de 32 caracteres
- **activityId**: Identificador único de actividad de 32 caracteres
- **name**: Nombre de la cookie
- **value**: Valor de la cookie (Si es superior a 255, se trunca)
- **expirationDate**: Timestamp de tipo long con la fecha de expiración de la cookie
- **url**: Dominio + path de la cookie

Cabe reseñar que las tablas de requests y trackers tienen una Foreign Key de la tabla de activity y ésta, a su vez, una Foreign Key sobre la tabla browsers.

Por último, se debe resaltar que en el software entregado, las tablas han sido creadas en un *schema* llamado "*privacyplugin*" en una base de datos local, por lo tanto la configuración del driver de *jdbc* se debe configurar teniendo en cuenta dicha parametrización.

Si se requiriese ejecutar en otra máquina o con un *schema* distinto, toda esta configuración se puede realizar a través del fichero *config.properties* situado en la carpeta de fuentes *src/main/resources*

config.properties

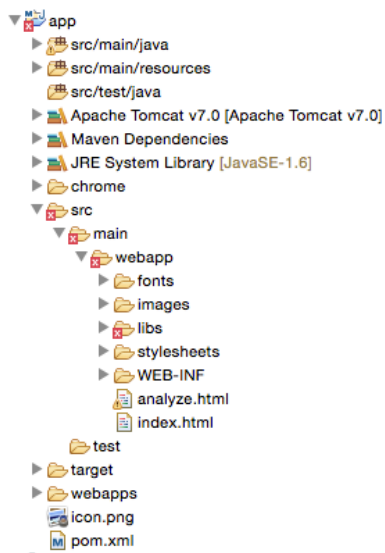
```
URL jdbc:mysql://%s/privacyplugin
HOST localhost
USER pfc
PASS uc3m
```

4.3. FRONT END

La parte front se ha implementado utilizando HTML5, JS y CSS3.

Como framework *js* sobre el que construir se ha elegido *dojo*, un framework muy potente patrocinado por IBM y sobre el cual el equipo de desarrollo ya tenía un conocimiento previo. Este punto ha sido clave para su elección.

La estructura de ficheros del front es la siguiente:



En ella podemos destacar las siguientes páginas:

4.3.1. Index

Se trata de la página principal del site y en ella se pueden realizar búsquedas de las páginas visitadas por un browser que haya tenido el plugin instalado.

Una vez mostradas todas las páginas visitadas, se podrán ver tanto las peticiones de recursos que dicha página solicitó, como las cookies que fueron creadas/modificadas durante dicha sesión.

Se explica con más detalle en el "ANEXO C: Documentación de usuario del Interfaz Web"

dcf8a0e1-907a-495e-ae2c-1457db7b7aa5

URL	Timestamp	Live
http://www.forocoches.com/	1444868705160	true
https://www.google.es/_/chrome/newtab?esprv=2&ie=UTF-8	1444868703542	true

Requests

URL	FrameId	ParentF...	Type
https://c.betrad.com/a/n/366/9595.js	28	19	script
http://www.forocoches.com/js/add2home_ipad.js	0	-1	script
https://googleads.g.doubleclick.net/pagead/ads?client=ca-pub-2852522843813544&output=html&h=90&slotname=3141121293&ad...	19	17	sub_frame
https://cm.g.doubleclick.net/pixel?google_nid=exp&google_push=AHNF13LzmH7sRRlQmncSI_DvclC...1y-X3Pu&google_ula=2786954&google_hm=18072662575594009220	25	14	image
https://bid.g.doubleclick.net/xbbe/creative/ad?d=APEucNVaopzG6bLjceep17bWXYVGZLgI5WLMMLJNNPKIndfpzi...Q8BWYnXyUk-edN5tO_D3WEGvURPy2LEMxe7Qib_SVvt_-3580n09YXFAS3Z0EG...w1V9g-MacVY	14	0	script
http://www.forocoches.com/js/add2home.css	0	-1	stylesheet
https://bs.serving-sys.com/BurstingPipe/adServer.bs?cn=rsb&c=2&pli=14764257&PluID=0&w=300&h=600&ncu=http://ad...	14	0	script

Trackers

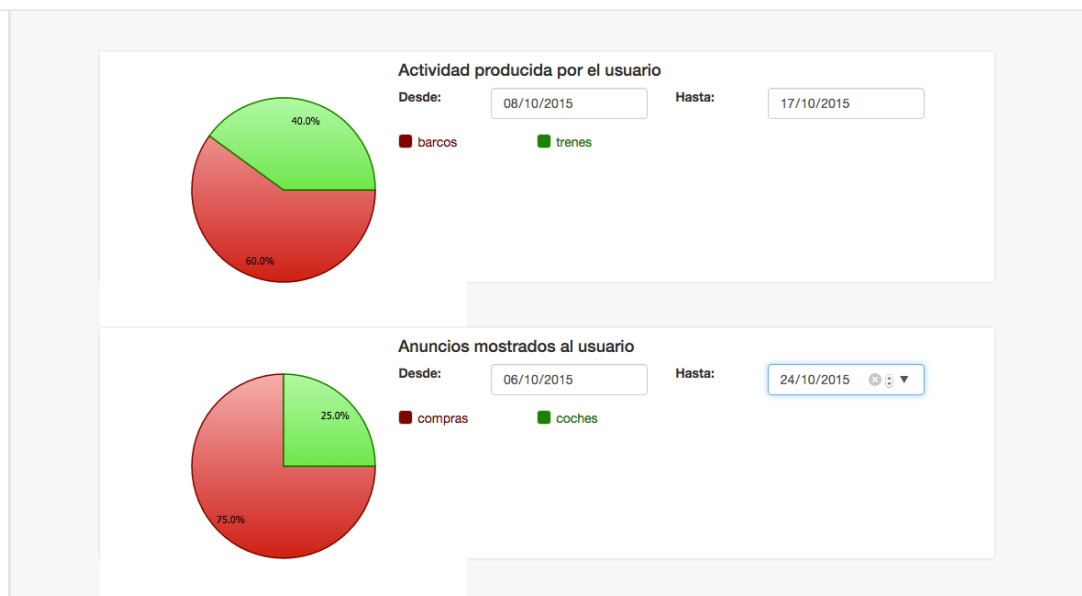
4.3.2. Análisis (Mock Up)

Aunque este requisito no formase parte de este proyecto (puesto que faltan varias piezas del ecosistema), se ha querido hacer un mockup para poder mostrar cómo se podría explotar la información recopilada.

En ella se propone que se elija un browser y, a continuación, se muestran dos áreas: una con la categorización de las visitas que se realizaron en un periodo de tiempo y otro área en el que se mostrarían las categorías de anuncios que le fueron mostrados.

Gracias al gráfico, se podría observar si la información se puede correlar y poder extrapolar que hay una relación causa/efecto entre visitas y anuncios.

dcf8a0e1-907a-495e-ae2c-1457db7b7aa5



5. PLANIFICACIÓN Y PRESUPUESTO

A continuación se expondrá una simulación de lo que hubiese supuesto el desarrollo de dicho proyecto si hubiese sido realizado por una empresa de consultoría de IT.

5.1. FASES DEL PROYECTO

Las fases del proyecto fueron las siguientes:

Toma de Requisitos

El consultor se reunió con el stakeholder del proyecto para recabar tanto los requisitos iniciales como el plan maestro del proyecto.

Durante esta reunión se estableció un plan de entregas negociable a futuro y una entrega inicial para contrastar si la solución ideada por el consultor cubría las expectativas del cliente.

A parte del tiempo de la reunión, en esta fase se engloba el análisis de la solución, y el diseño inicial de la propuesta.

Iteración 1

En ella se presenta el primer conjunto de historias de usuario. Se trata de realizarlo en un corto espacio de tiempo que permita corroborar que lo que propone el cliente y lo que idea el consultor está totalmente alineado.

Iteración 2

La segunda iteración es más larga en tiempo que la anterior, pues una vez contrastada la solución es necesario el asentar las bases para tener una base sólida sobre lo que desarrollar el futuro proyecto.

Así mismo, las historias de usuario que entran en dicha iteración son de mayor entidad y, por consiguiente, con mayor dificultad técnica.

Iteración 3

La tercera iteración es, si cabe, algo más larga. Sus historias de usuario son igual de complejas pero, ya que el proyecto está a punto de finalizar, se deben ir cerrando la funcionalidad de diversos módulos que parece que no van a seguir evolucionando.

Iteración 4

Se trata de la última iteración y en la que se prepara la entrega realizando la documentación necesaria para poder entregar el producto final

Entrega Final

La defensa del proyecto tiene entidad propia y merece ser tenida en cuenta como la última fase del proyecto en el que el trabajo realizado es valorado por el cliente (en este caso el tribunal) y es donde se juzga el éxito o fracaso del proyecto.

5.2. PERSONAL

El proyecto se ha llevado a cabo en modo ágil, iterativo e incremental, desarrollado enteramente por una única persona con un nivel medio/alto en este tipo de desafíos, capaz de tomar requisitos, diseñar la solución, implementarla y documentarla.

Las habilidades técnicas del desarrollador deben estar orientadas tanto al desarrollo de front (conocimientos de HTML, CSS y, sobre todo, javascript) como al desarrollo de back (conocimientos de Java y base de datos)

También será necesario tener la habilidad de empatizar con el stakeholder del proyecto y poder sacar los distintos requisitos del proyecto, así como la capacidad de dividirlo adecuadamente en historias de usuario que se vean reflejadas en distintas entregas a lo larga de diferentes iteraciones.

5.3. ESTIMACIÓN DE TIEMPOS Y COSTES

A continuación se realiza una estimación de tiempos y costes para la realización del proyecto.

La planificación se realiza en base a jornadas de 8 horas.

Fase	Días	Horas	Coste hora	Coste total
Toma de requisitos inicial y análisis	2	16	35	560
Iteración 1	3	24	35	840
Iteración 2	7	56	35	1960
Iteración 3	10	80	35	2800
Iteración 4	5	40	35	1400
Entrega final	0,5	4	35	140
TOTAL	27,5	220		7700

El coste del proyecto se estima en unos 7.700 euros.

El coste de material (equipos, conexión a internet, teléfono) está englobado en los 35 euros/hora y, dado que las reuniones con el stakeholder y la presentación final se realiza dentro de la misma Comunidad Autónoma y ambas no duran más de media jornada, no da lugar al cómputo de dietas.

6. CONCLUSIONES Y FUTUROS DESARROLLOS

A continuación se exponen tanto las conclusiones a las que se han llegado durante el desarrollo del proyecto como un listado de futuros desarrollos para el montaje del ecosistema final.

Como último punto se detallan las dificultades presentadas durante el proyecto

6.1. CONCLUSIONES

Las conclusiones de dicho proyecto deben dividirse entre conclusiones éticas, que serían aquellas conclusiones etéreas que pretenden alertar sobre la importancia de la difusión de datos personales en la red y conclusiones tangibles, que serían las entregas de software realizadas y que serán una pieza importante en un montaje mucho más complejo que permita denunciar los posibles abusos que se cometan en la red.

Conclusiones éticas

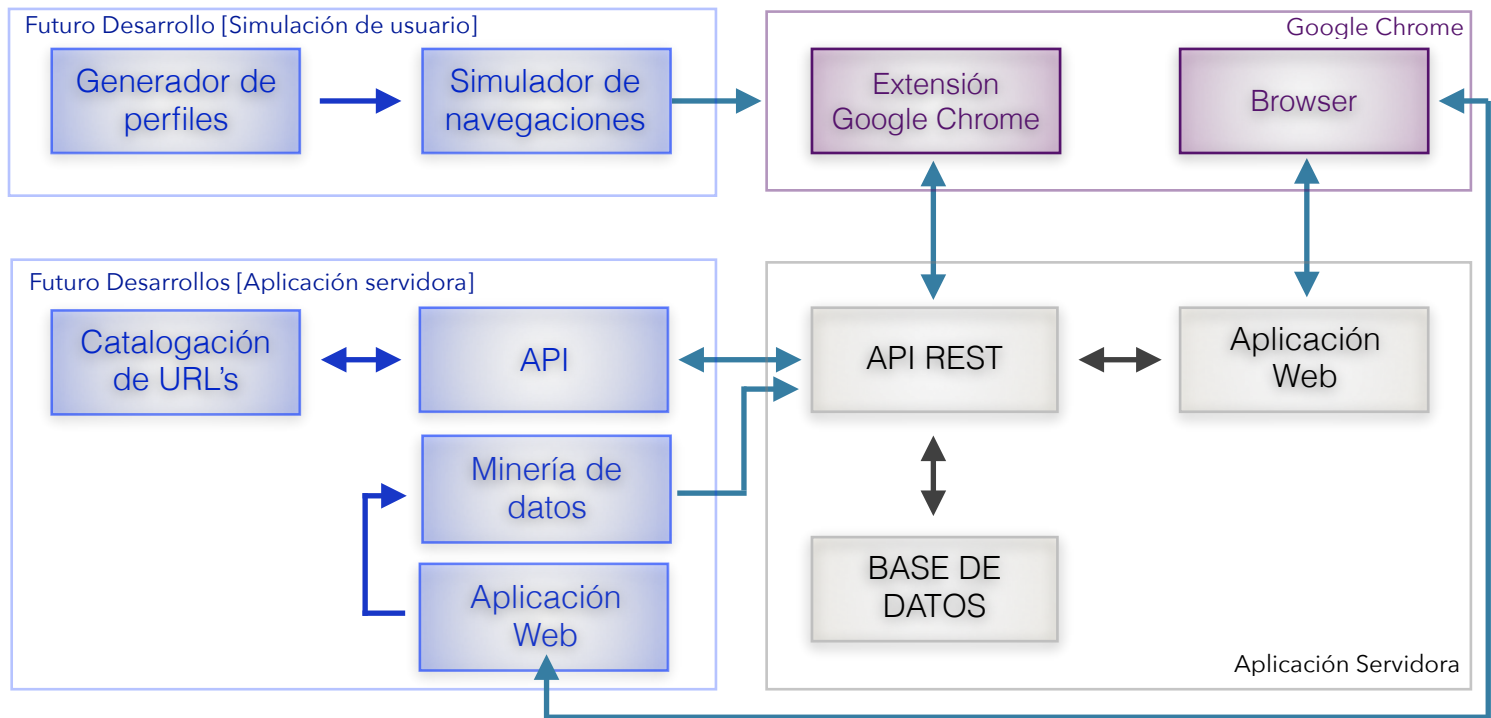
- Internet es una fuente de contenido, entretenimiento, comunicación,.. sin la que no se podría entender la sociedad actual. A cambio las libertades del usuario se ven recortadas pues su derecho a la privacidad se ve gravemente perjudicada.

Conclusiones tangibles

- **Plugin de Google Chrome:** Se trata de un fichero con extension .crx, que es el empaquetado de la extensión desarrollada y que permite ser instalada en un Google Chrome.
- **war instalable en un contenedor de aplicaciones java:** Se trata del empaquetado que contiene la aplicación que se ha desarrollado. En ella se incluye tanto la aplicación web como el API REST que permite obtener y guardar los datos en el servidor.

6.2. FUTUROS DESARROLLOS

A continuación se propone un posible esquema para el futuro montaje del ecosistema final.



Simulación de usuario

Sería interesante el desarrollo de un módulo que generase perfiles de usuario y simulase navegaciones sobre un browser que tenga instalado el plugin que se ha realizado.

El submódulo de generación de perfiles de este nuevo elemento debería permitir seleccionar un tipo de usuario con diversas características como podrían ser su edad, aficiones, condición sexual, creencias religiosas, motivaciones políticas, enfermedades, .. y debería producir un perfil de usuario y un calendario asociado en el que se agendasen qué días y a qué horas dicho usuario visitaría determinadas páginas.

De este modo, otro submódulo se nutriría de dicho calendario y, en el día adecuado a la hora prevista, instrumentalizaría al browser para realizar las visitas a las url's seleccionadas.

Así se podría entrenar un sistema para ejecutar visitas, de forma totalmente desatendida, y poder evaluar si los anuncios que se muestran a continuación tienen relación con el tipo de perfil creado.

Nuevos Módulos de la Parte Servidora

Se propone crear otro nuevo módulo en la parte servidora que permita realizar minería de datos con la información recolectada por el plugin. Este nuevo módulo podría tener acceso directo a la Base de Datos o explotarla mediante el API REST que provee el módulo actual. Una vez obtenido y tratados los datos, se debe dotar al nuevo módulo de un interfaz web que permitiese mostrar al usuario los resultados obtenidos.

Por otro lado, también se debería desarrollar un módulo que permitiese obtener, dada una URL, en qué categoría se engloba. De esta manera la aplicación actual podría no solo guardar qué url's visita el usuario si no a qué categoría pertenecen.

A parte de poder devolver la categorización de una página, este módulo debería también devolver la categorización de un anuncio para poder comparar a posteriori la correlación visitas/anuncios.

Este servicio de categorización debería estar expuesto a través de un API REST para aislarlo de la aplicación actual o podría estar embebido en la misma.

Considerando un escenario en el que no se trate de una aplicación monolítica compuesta de módulos que se instalan en la misma máquina, o en clusters balanceando la carga, se presenta la alternativa de los microservicios, donde el escalado no tendría porqué ser lineal.

En este escenario, tendría sentido escalar los módulos de recolección de datos y el que realiza la minería de datos dependiendo del número de usuarios activos, mientras que el servicio de categorización podría mantenerse más acotado.

6.3. DIFICULTADES ENCONTRADAS

El principal reto ha sido la creación del plugin, que es la piedra angular del desarrollo. Está basado en el API que expone Google Chrome y, por lo tanto, solo se pueden desarrollar las funcionalidades que dicho API soporta.

De este modo, no se han podido desarrollar características inicialmente requisitadas como la posibilidad de saber cuando un plugin es desinstalado (ya que el API de Chrome no da esta posibilidad)

Otro reto ha sido la recolección y envío de las url's por parte del plugin y la actualización de dicha información en la base de datos. Al final se optó por utilizar el local storage del browser como una caché que permitiese ir guardando toda la información para evitar que se perdiese ante cierres voluntarios/involuntarios del navegador.

Como último punto, cabe reseñar que ningún desarrollo está exento de que surjan problemas inesperados. En este, un par de meses antes de la entrega del proyecto y cuando el software estaba completamente desarrollado, la maquina donde estaba siendo desarrollado comenzó a ir lento, más lento, hasta que un día no arrancó. El servicio técnico de Apple constató que se había roto el disco duro y que era imposible recuperar la información del mismo. Por suerte (o por precaución) se habían hecho distintas copias de seguridad que permitieron recuperar el trabajo y no darlo por perdido.

7. REFERENCIAS

A continuación se exponen una serie de términos y de referencias que se han consultado durante el desarrollo del proyecto.

7.1. DICCIONARIO DE TÉRMINOS

Activity: En la memoria, el termino Activity o Actividad se refiere a aquella actividad que ha sido monitorizada por el Browser (que tiene instalado el plugin de privacidad) y que es grabada en la Base de Datos.

Una Actividad se corresponde con una visita a una página web. Esta visita se podrá haber producido durante una monitorización activa del plugin o tratarse de una entrada recolectada desde el historial del navegador. Este hecho se diferencia a través del flag "live".

Dado que una Actividad siempre se corresponde con un Browser, la actividad tiene como Foreign Key el identificador del browser con el que está relacionado.

A su vez, una visita a una página tiene aparejado un listado de recursos/urls que dicha página descarga y que se denominan Requests y un listado de Trackers (cookies). Por ello, tanto los Requests como las Trackers tienen una Foreign Key contra el identificador de Actividad.

Ágil (o agile): El desarrollo ágil de software envuelve un nuevo enfoque radical para la toma de decisiones en los proyectos de software, refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto, así el trabajo es realizado mediante la colaboración de equipos auto-organizados y multidisciplinarios, inmersos en un proceso de toma de decisiones a corto plazo compartido.

https://es.wikipedia.org/wiki/Desarrollo_Ágil_de_software

<http://www.agilemanifesto.org/iso/es/>

Browser: Normalmente un browser se corresponde con la definición de “navegador web”. En la memoria de este proyecto se asimila a aquel navegador web que tiene instalado el plugin de privacidad y el que, por lo tanto, podrá ser indexado en la tabla de la base de datos mediante tres valores: Un identificador único de 32 caracteres, un timestamp de creación y uno de modificación.

Dado que sobre el Browser se recolectarán las urls visitadas, el identificador único de un browser será una Foreign Key para la tabla de Activity (ver Activity)

Cookies: Una cookie es una pequeña información enviada por un sitio web y almacenada en el navegador del usuario, de manera que el sitio web puede consultar la actividad previa del usuario.

Sus principales funciones son:

- Llevar el control de usuarios: cuando un usuario introduce su nombre de usuario y contraseña, se almacena una cookie para que no tenga que estar introduciéndolas para cada página del servidor. Sin embargo, una cookie no identifica a una persona, sino a una combinación de computador-navegador-usuario.
- Conseguir información sobre los hábitos de navegación del usuario, e intentos de spyware (programas espía), por parte de agencias de publicidad y otros. Esto puede causar problemas de privacidad y es una de las razones por la que las cookies tienen sus detractores.

[https://es.wikipedia.org/wiki/Cookie_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Cookie_(inform%C3%A1tica))

Foreign Key (Clave foránea): En el contexto de bases de datos relacionales, una clave foránea o clave ajena (o Foreign Key FK) es una limitación referencial entre dos tablas. La clave foránea identifica una columna o grupo de columnas en una tabla (tabla hija o referendo) que se refiere a una columna o grupo de columnas en otra tabla (tabla maestra o referenciada). Las columnas en la tabla referendo deben ser la clave primaria u otra clave candidata en la tabla referenciada.

https://es.wikipedia.org/wiki/Clave_for%C3%A1nea

Hardcode: Término del mundo de la informática hace referencia a una mala práctica en el desarrollo de software que consiste en incrustar datos directamente (a fuego) en el código fuente del programa, en lugar de obtener esos datos de una fuente externa como un fichero de configuración o parámetros de la línea de comandos, o un archivo de recursos.

https://es.wikipedia.org/wiki/Hard_code

Historia de usuario: Es una representación de un requisito escrito en una o dos frases utilizando el lenguaje común del usuario. Las historias de usuario son utilizadas en las metodologías de desarrollo ágiles para la especificación de requisitos (acompañadas de las discusiones con los usuarios y las pruebas de validación). Cada historia de usuario debe ser limitada, ésta debería poderse escribir sobre una nota adhesiva pequeña. Dentro de la metodología XP las historias de usuario deben ser escritas por los clientes.

https://es.wikipedia.org/wiki/Historias_de_usuario

[https://en.wikipedia.org/wiki/INVEST_\(mnemonic\)](https://en.wikipedia.org/wiki/INVEST_(mnemonic))

Mock up: Es un modelo utilizado para la demostración, evaluación del diseño, promoción, y para otros fines. Un mockup es un prototipo si proporciona al menos una parte de la funcionalidad de un sistema y permite pruebas del diseño. Los mock up son utilizados por los diseñadores principalmente para la adquisición de comentarios por parte de los usuarios.

<https://es.wikipedia.org/wiki/Mockup>

Request: En la memoria, Request se refiere a una url o a un recurso descargado durante la visita de una página, pudiendo ser este una imagen, un js, un iframe,..

Este elemento podrá ser grabado en la base de datos y mantendrá una referencia en forma de Foreign Key contra el identificador de una determinada Activity (es decir, a la visita a la página durante la cual fue recolectada)

Schema: Describe la estructura de una base de datos, en un lenguaje formal soportado por un sistema de gestión de base de datos (DBMS). En una base de datos relacional, el esquema define sus tablas, sus campos en cada tabla y las relaciones entre cada campo y cada tabla.

https://es.wikipedia.org/wiki/Esquema_de_una_base_de_datos

Stakeholder: Su significado se refiere a que en las organizaciones participan diversos grupos responsables además de sus propietarios. Dichos grupos son todas las personas, organizaciones y empresas que tienen interés en una empresa u organización dada. Ejemplos de stakeholders de una compañía serían: los empleados, los clientes, los proveedores de bienes y servicios, los proveedores de capital, la comunidad, y la sociedad.

<https://es.wikipedia.org/wiki/Stakeholder>

Tracker: En la memoria, Tracker se refiere al modelo que se guarda en la base de datos y que modela a una cookie de una página web.

El Tracker tendrá como Foreign Key el identificador de la Actividad en la que fue recolectada.

7.2. BIBLIOGRAFÍA

A continuación se presentan una relación de libros recomendables para la realización del presente proyecto:

Mastering Apache Maven 3

Prabath Siriwardena

Packt Publishing

RESTful Java with JAX-RS

Bill Burke

O'Reilly

MySQL Pocket Reference

George Reese

O'Reilly

Developing RESTful Services with JAX-RS 2.0, WebSockets and JSON

Masoud Kalali & Bhakti Mehta

Packt Publishing

**Getting Started with Grunt: The
JavaScript Task Runner**

Jaime Pillora

Packt Publishing

The Agile Samurai

Jonathan Rasmusson

The Pragmatic Programmers

7.3. SITES

Durante la realización del proyecto y de ésta memoria, se ha necesitado buscar documentación tanto técnica como divulgativa. Estos han sido sólo algunos de los sitios consultados.

Agencia Española de Protección de Datos

<https://www.agpd.es>

Instituto de Estadística de Cataluña

<http://www.idescat.cat/economia/inec?tc=3&id=8707&lang=es>

Dojo

<http://dojotoolkit.org>

Maven

<https://maven.apache.org/>

Jersey

<https://jersey.java.net/>

Extensiones de Google Chrome

<https://developer.chrome.com/extensions>

<https://developer.chrome.com/extensions/devguide>

https://developer.chrome.com/extensions/api_index

7.4. ARTÍCULOS

"Tracking Pixel, qué es y cómo se utiliza"

<http://www.miltonfactory.com/blog/index.php/performance-marketing/tracking-pixel-que-es-y-como-se-utiliza-en-publicidad-online/>

"¿Eres consciente de la privacidad en internet?"

<http://www.abc.es/tecnologia/redes/20150128/abci-proteccion-datos-personas-wearables-201501271726.html>

"España aumenta el número de usuarios activos en redes sociales en 2014 y llega a los 17 millones"

<http://www.obs-edu.com/noticias/estudio-obs/espana-aumenta-el-numero-de-usuarios-activos-en-redes-sociales-en-2014-y-llega-los-17-millones/>

"Mantener mi privacidad en google"

<http://www.portalprogramas.com/milbits/informatica/mantener-mi-privacidad-en-google.html>

ANEXO A - Manual de instalación de la extensión

Las extensiones de Google Chrome pueden ejecutarse en dos modalidades distintas:

- **Extensión descomprimida:** Util cuando la extensión está en fase de desarrollo. Se selecciona la carpeta donde se tienen los fuentes del plugin y cualquier cambio realizado en el fuente es tomado en caliente sin necesidad de empaquetarlo.
- **Extensión empaquetada:** Instalación final que realiza el usuario final. Se trata de un único fichero con extensión `.crx` que se genera a partir de los fuentes y que mantiene la estructura de carpetas del plugin comprimida y ejecutable

Modo Descomprimido

Para poder hacer uso de un plugin en modo descomprimido se debe ir al gestor de extensiones de Chrome a través del menú o tecleando la url:

`"chrome://extensions/"`

Una vez en el gestor, se debe marcar la opción "Modo de desarrollador"



Y aparecerán las siguientes opciones:



Seleccionando el botón "Cargar extensión descomprimida" aparecerá una ventana que solicitará la carpeta de la extensión. Una vez seleccionada, la extensión aparecerá como sigue:

Extensiones

☒ Modo de desarrollador

Cargar extensión descomprimida...

Empaquetar extensión...

Actualizar extensiones ahora



Privacy Plugin 1.0

Privacy Plugin..

[Permisos](#) [Volver a cargar \(%R\)](#)

ID: plelelfpifodlnegmigdmjmfjnaiecn

Cargado desde:

[~/Documentos/eclipse/plugin/repos/src/main/resources/extensions/chrome](#)

☐ Habilitar



Pulsando la papelera la extensión será borrada (La carpeta original no será afectada)

Cuando se habilita la opción "Habilitar" la extensión se vuelve activa.

Extensiones

☒ Modo de desarrollador

Cargar extensión descomprimida...

Empaquetar extensión...

Actualizar extensiones ahora



Privacy Plugin 1.0

Privacy Plugin..

[Permisos](#) [Volver a cargar \(%R\)](#)

ID: plelelfpifodlnegmigdmjmfjnaiecn

Cargado desde:

[~/Documentos/eclipse/plugin/repos/src/main/resources/extensions/chrome](#)

Inspeccionar vistas: [página en segundo plano](#)

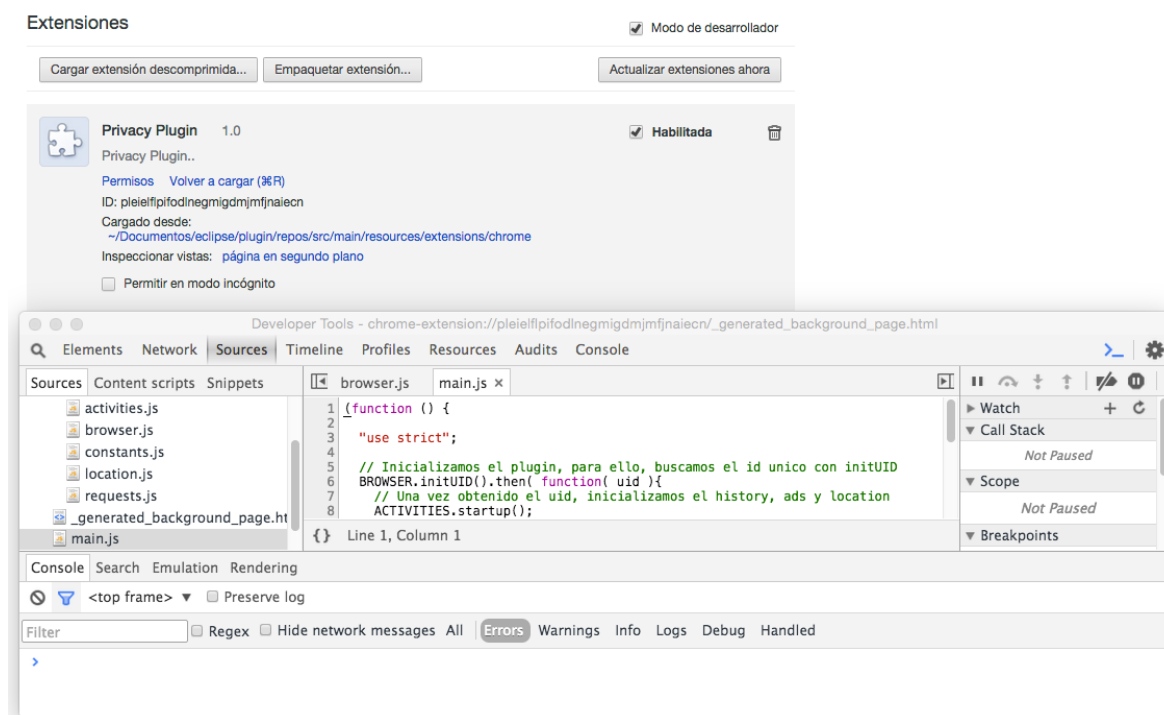
☐ Permitir en modo incógnito

☒ **Habilitada**



Si se pulsa "Volver a cargar" la extensión se reiniciará.

Si se pulsa el enlace “página en segundo plano” aparecerá el debugger donde se podrá ver el código, permitiendo tareas propias del debugging como pueden ser poner puntos de extensión sobre el código fuente o ver la consola.

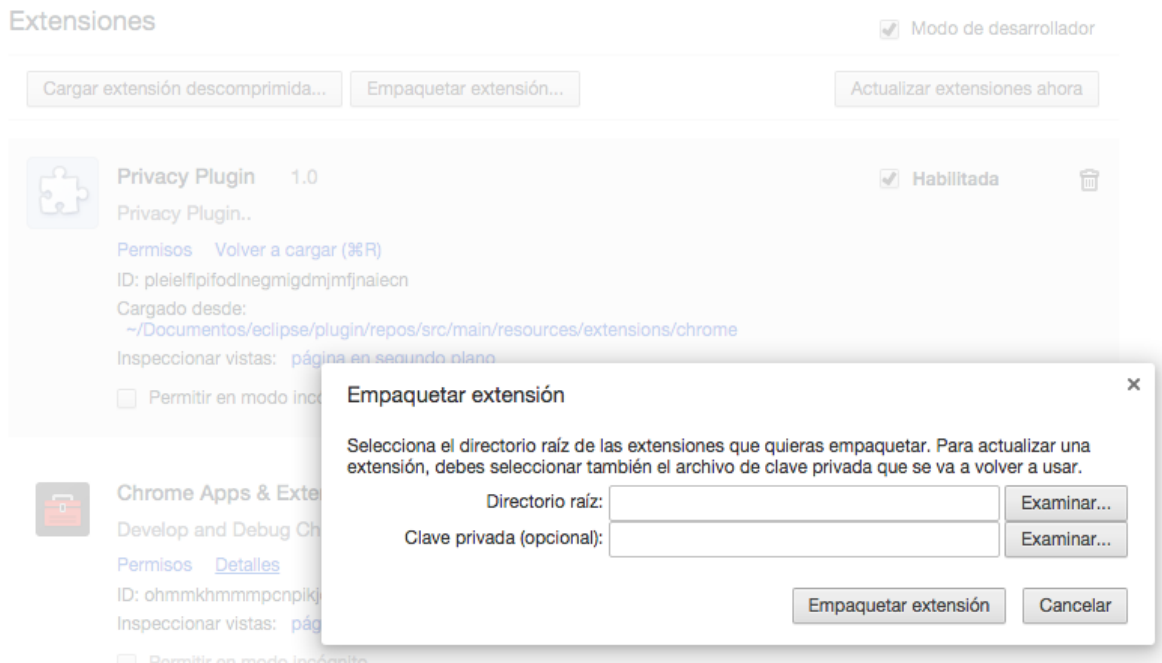


Modo empaquetado

Si se desea instalar en modo empaquetado, lo primero que se necesita es crear el empaquetado, para ello, desde el mismo gestor de extensiones, teniendo activado el “modo desarrollador” se deberá pulsar la opción de “Empaquetar extensión..”

Una vez hecho esto aparecerá una ventana solicitando la carpeta del plugin y una clave privada (Por si se quiere firmar el plugin y poder crear nuevas versiones del mismo)

Una vez pulsado “Empaquetar extensión” el gestor indicará que se ha generado con éxito la extensión y en qué ruta se encuentra



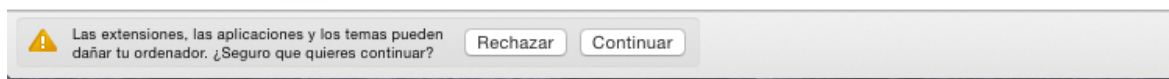
Viendo la ruta indicada, se habrán generado los siguientes ficheros:

- **chrome.crx:** Es la extensión propiamente dicha
- **chrome.pem:** Es la clave que habrá que incluir en la anterior ventana para poder crear nuevas versiones de éste plugin.

Una vez hecho esto, ya se podría subir el plugin al market de Chrome o instalarlo de manera local en la máquina. Si se quiere hacer esto segundo, deberemos marcar al browser que queremos instalar plugins de terceros abriendo el browser de un modo especial siguientes los siguientes pasos:

1. Dirigirse al acceso directo del navegador. Una vez allí, hacer clic en el botón derecho y seleccionar Propiedades.
2. En la pestaña “Acceso directo” ir al cuadro Destino.
3. Al final del código que aparece poner lo siguiente:
--enable-easy-off-store-extension-install

Una vez hecho esto, instalar, se debe hacer doble click sobre la extension o arrastrarla sobre el gestor de extensiones y continuar con la instalación:



Pulsando continuar, la extensión se habrá instalado correctamente.

ANEXO B - Manual de instalación de la aplicación servidora

La aplicación servidora tiene dos módulos distintos, una base de datos (MySQL) y una aplicación Java que se conecta a dicha base de datos.

A continuación se expondrán los pasos a realizar para ejecutar la aplicación servidora desarrollada.

Aplicación Java

La aplicación servidora se sirve de maven para obtener sus dependencias y construir, como producto final, un war que podrá ser instalado en un contenedor de aplicaciones (en este caso, un Apache Tomcat v7).

La aplicación se ha desarrollado en eclipse (se ha utilizado la versión de “Spring Tool Suite 3.7.0”, aunque no es necesario utilizar la misma)

Como primer paso, se deberá descargar una versión de dicho IDE. Para ello, se puede ir a la siguiente URL:

<http://www.eclipse.org/downloads/>

Mac OS X users please note: Eclipse requires Mac OS X 10.5 (Leopard) or greater. You might get a warning on some installations due to recent changes how Apple treats signatures. Please see [this post](#) for a possible solution.

Eclipse Mars.1 (4.5.1) Release for Mac OS X (Cocoa)

Try the Eclipse Installer **NEW**

The easiest way to install and update your Eclipse Development Environment.

FIND OUT MORE

Mac OS X
64 bit

Windows
32 bit | 64 bit

Linux
32 bit | 64 bit

Optimizer
by ZERO TURNAROUND
for Eclipse
FREE TOOL

RELATED LINKS

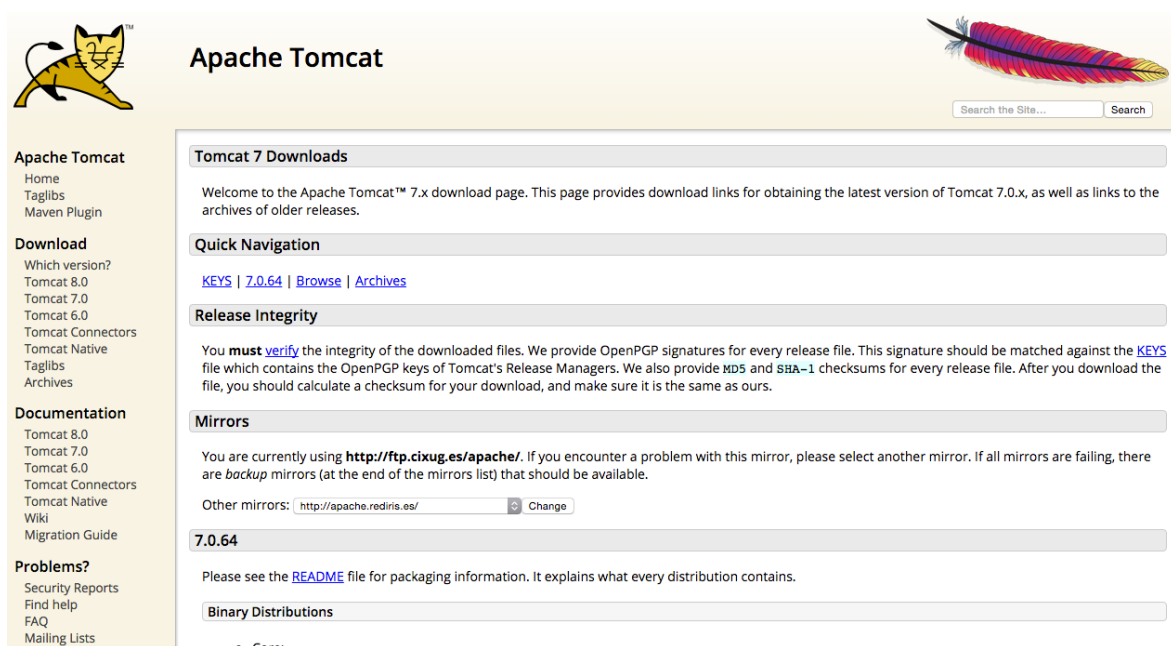
- Compare & Combine Packages
- New and Noteworthy
- Install Guide

Nota: Se recomienda utilizar una versión que incluya todo lo necesario (Eclipse M2E, Adaptador de Apache) para el desarrollo de aplicaciones web. De lo contrario, se tendrán que descargar dichas funcionalidades vía Eclipse Marketplace.

Una vez descargado, se deberá extraer en cualquier carpeta y hacer doble click sobre eclipse.ex para ejecutar la aplicación. El IDE de Eclipse no necesita de instalación previa, con lo que se podrá mover de una carpeta a otra, o ser renombrada sin problemas.

Una vez extraído y arrancado, se deberá configurar un servidor tomcat donde ejecutar la aplicación. Para ello, como primer paso, será necesario descargar una versión de dicho servidor desde el siguiente enlace:

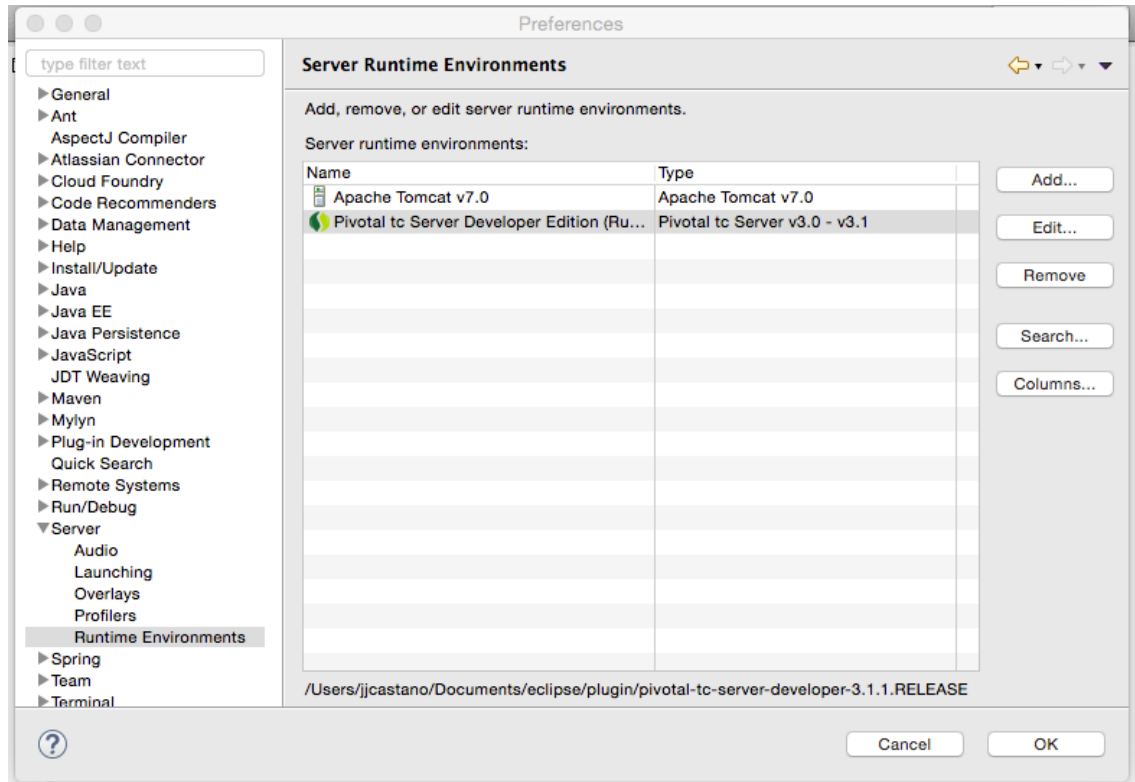
<http://tomcat.apache.org/download-70.cgi>



NOTA: La version con la que se ha certificado el software ha sido Tomcat 7

Para poder ser utilizado, se deberá descomprimir el zip descargado y recordar la ruta de destino ya que se deberá configurar Eclipse con dicha ruta.

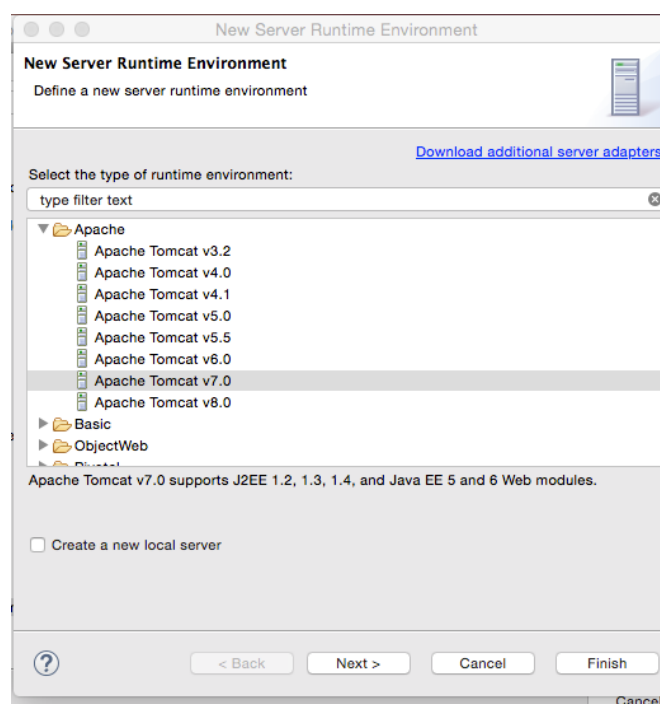
Para ello, se deberá ir abrir la página de preferencias de eclipse (windows: Ventana > Preferencias; Mac: Eclipse > Preferencias)



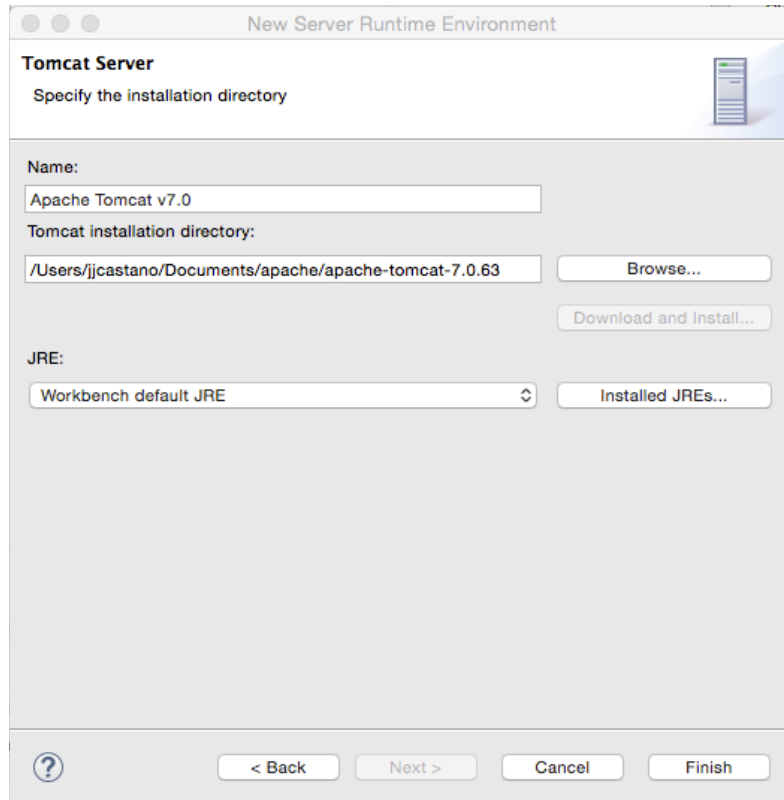
En el árbol de la izquierda se deberá seleccionar la ruta:

“Server > Runtime Environments”

.. y seleccionar el botón “Add”. Hecho esto aparecerá el siguiente cuadro de diálogo:

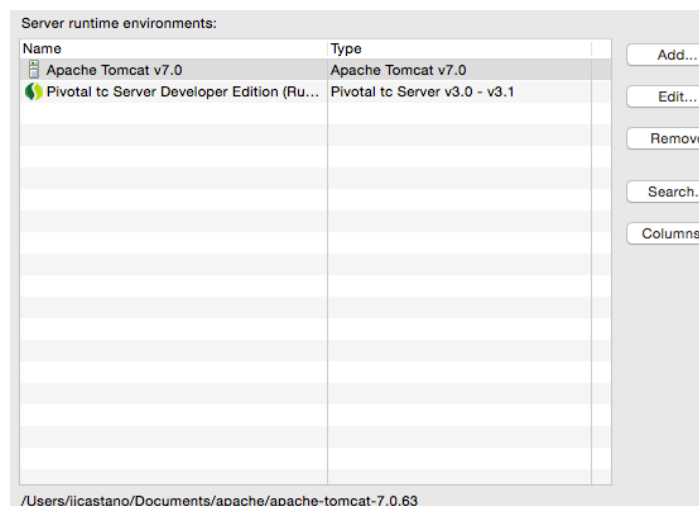


Se debe seleccionar "Apache > Apache Tomcat v7.0" y pulsar "Next >". Se mostrará la siguiente pantalla:

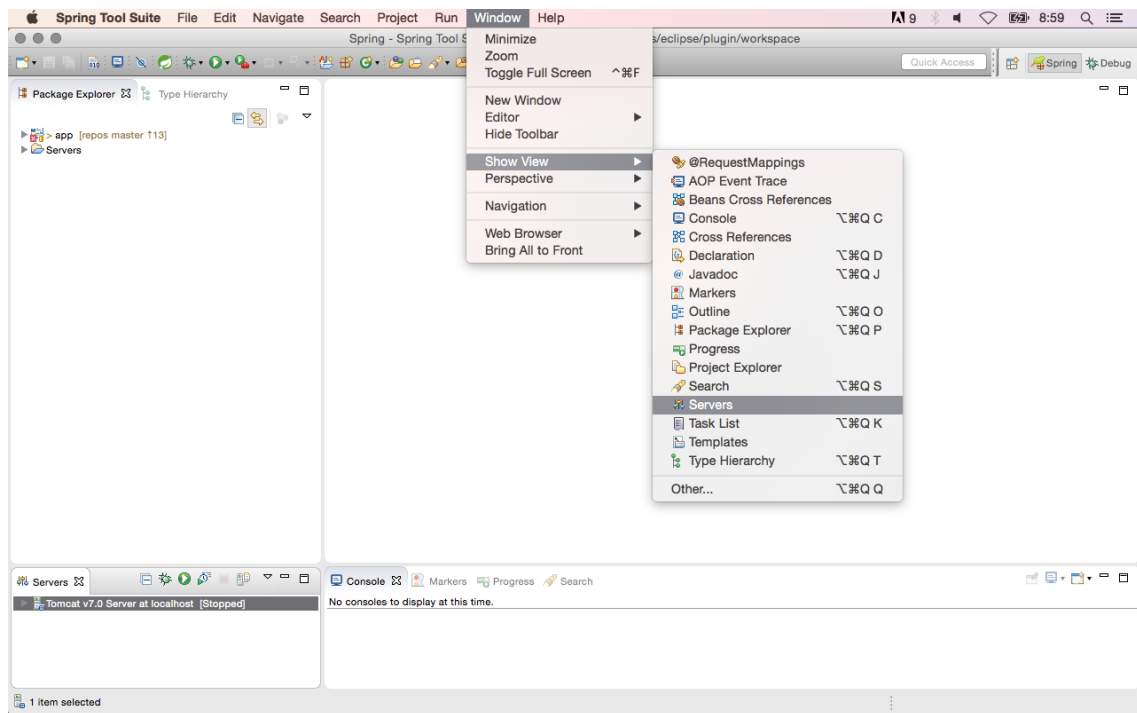


En "Tomcat installation directory" se deberá poner la ruta donde se extrajo el zip de tomcat y pulsar "Finish" para terminar de configurar.

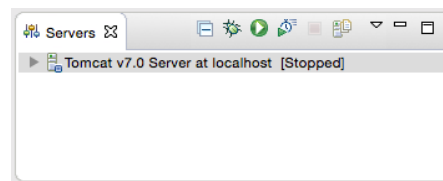
Una vez hecho esto, la tabla de servidores de la ventana de Preferencias deberá mostrar la configuración de tomcat como sigue:



Pulsando "OK" y yendo a la vista de Servidores de Eclipse:

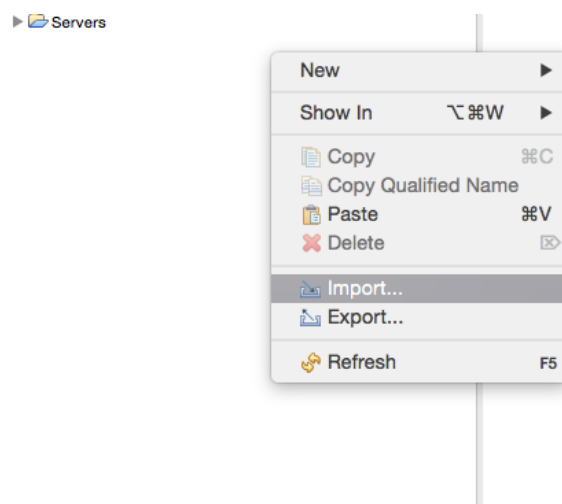


Se mostrará la siguiente vista:

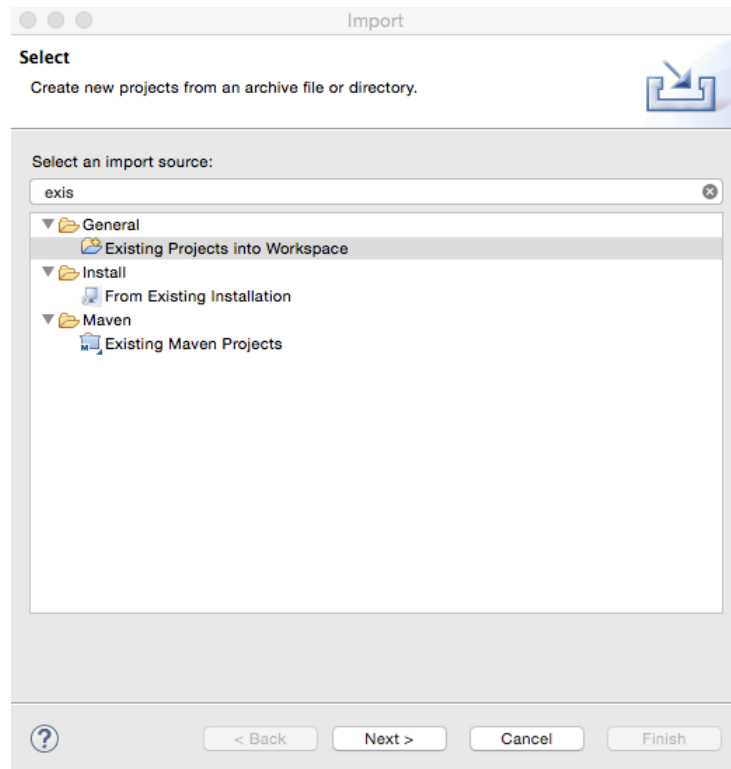


Una vez configurado el servidor, se deberá proceder a importar el proyecto Java contenido en el fichero app.zip adjunto.

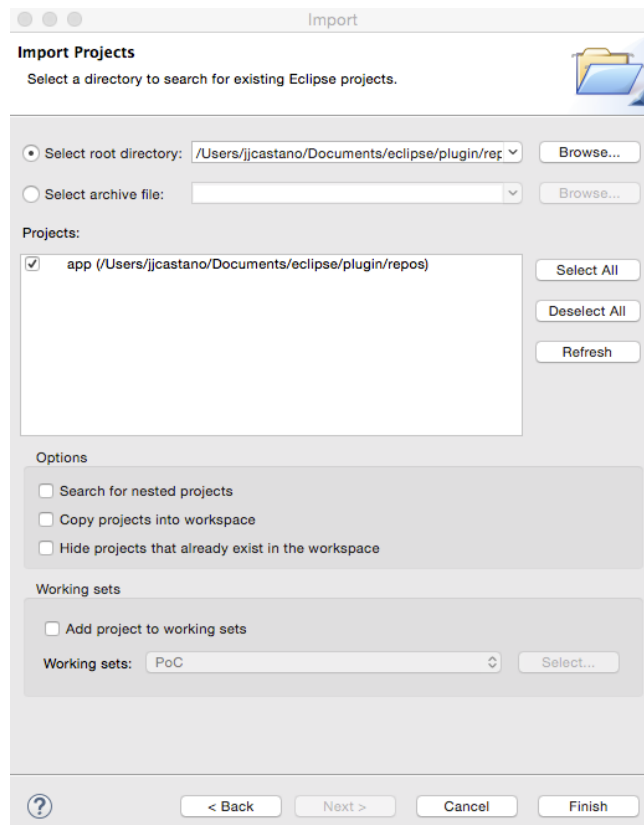
Para ello, se deberá extraer el zip en una carpeta, recordar la ruta e importarlo haciendo click derecho en la vista "Package Explorer" y seleccionando "Import..."



Posteriormente se debe seleccionar "Existing Projects into Workspace" y pulsar "Next >"

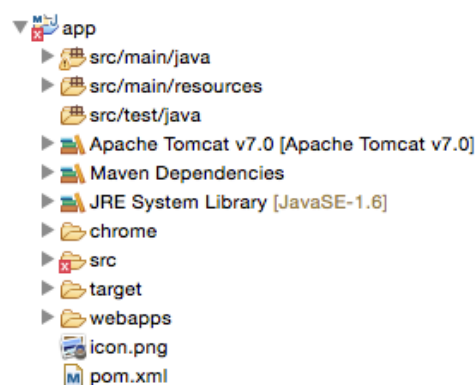


Una vez aparece el wizard de importación..

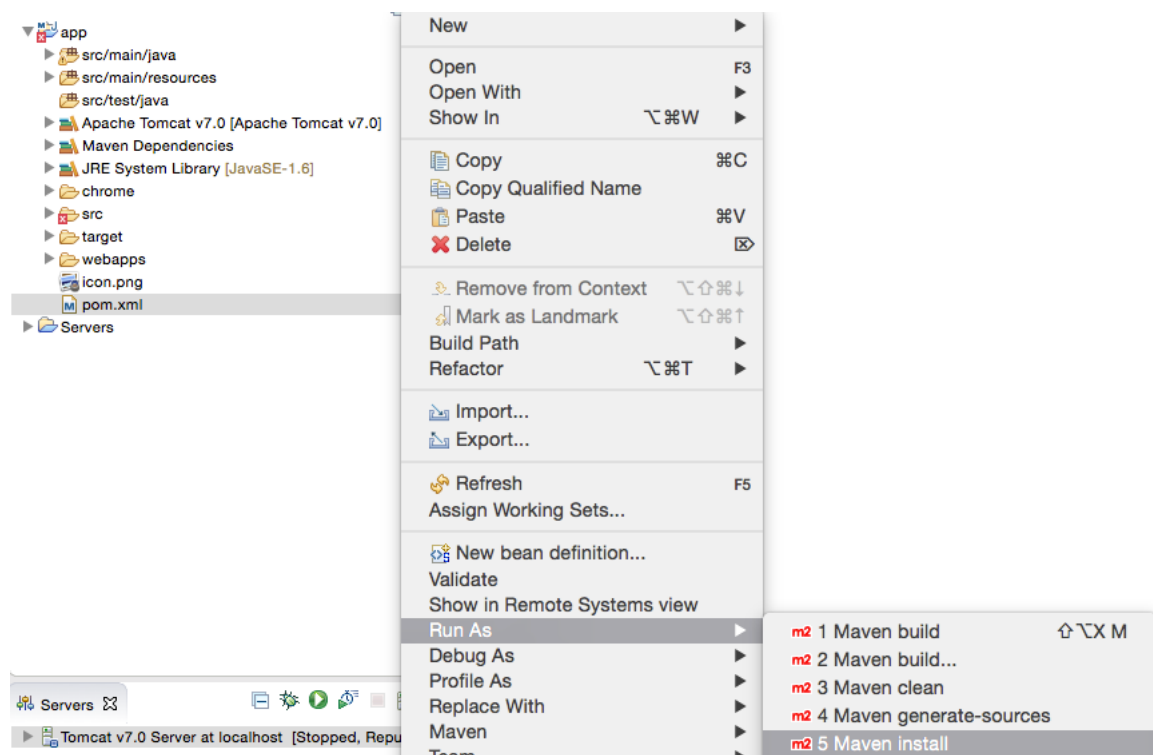


Se debe buscar la ruta donde se extrajo el .zip y ponerlo en "Select root directory". Una vez hecho esto, en la lista "Projects" aparecerá "app", que es el nombre del proyecto que se deberá importar. Se debe seleccionar dicho proyecto y, dependiendo de si se quiere copiarlo al workspace o mantenerlo en la carpeta existente, se deberá pulsar el check "Copy projects into workspace".

Pulsando finalizar, el proyecto aparecerá en el árbol de "Package Explorer". La aplicación ya estará importada y lista para ser construida.



Para ello, es necesario seleccionar el fichero pom.xml de la raíz del proyecto, hacer click derecho sobre él y lanzar "Run As > Maven install".

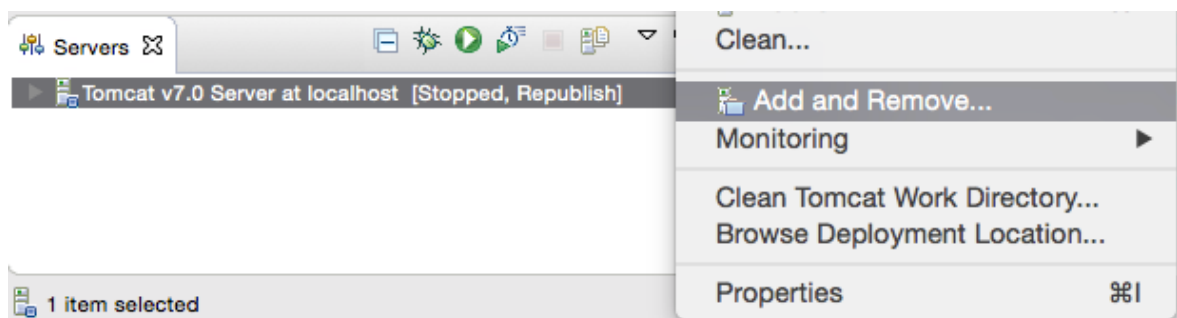


Esta acción descargará todas las dependencias del proyecto y realizará la construcción maven descrita en dicho fichero. El resultado saldrá por la consola de Eclipse de la siguiente manera:

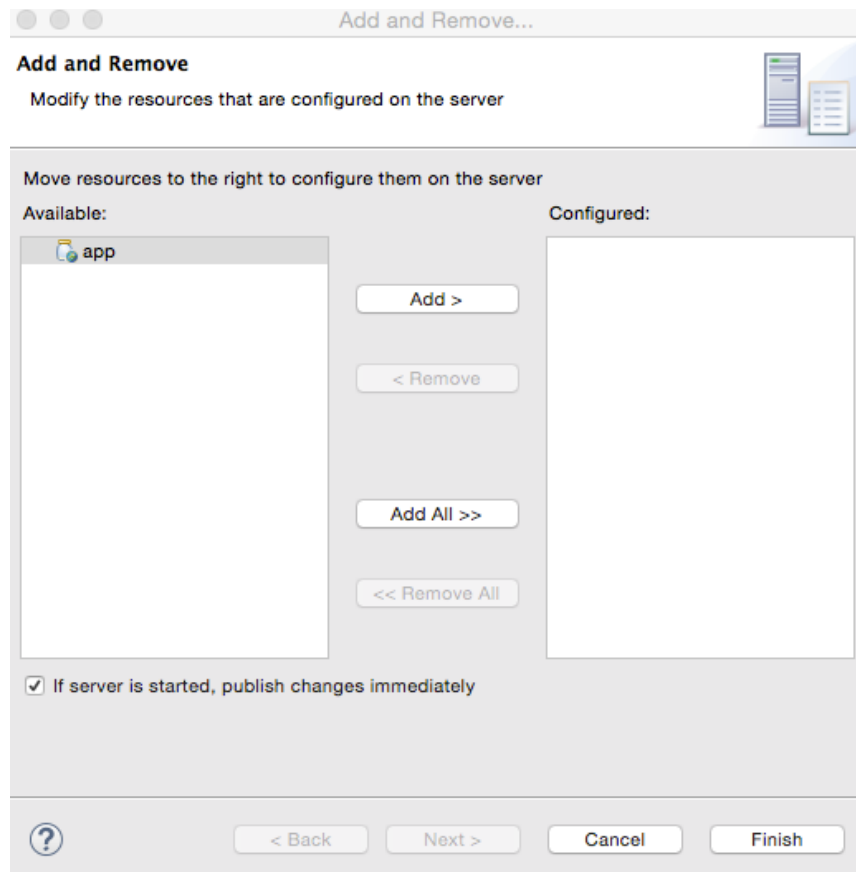
```
Console  [Markers] [Progress] [Search]
<terminated> /Library/Java/JavaVirtualMachines/jdk1.8.0_51.jdk/Contents/Home/bin/java (15 de oct. de 2015 22:49:34)
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building Privacy Plugin 1.0
[INFO] -----
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ privacy-plugin ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 2 resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ privacy-plugin ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 41 source files to /Users/jjcastano/Documents/eclipse/plugin/repos/target/classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ privacy-plugin ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /Users/jjcastano/Documents/eclipse/plugin/repos/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ privacy-plugin ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ privacy-plugin ---
[INFO]
[INFO] --- maven-war-plugin:2.1.1:war (default-war) @ privacy-plugin ---
[INFO] Packaging webapp
[INFO] Assembling webapp [privacy-plugin] in [/Users/jjcastano/Documents/eclipse/plugin/repos/target/privacy-plugin-1.0]
[INFO] Processing war project
[INFO] Copying webapp resources [/Users/jjcastano/Documents/eclipse/plugin/repos/src/main/webapp]
[INFO] Webapp assembled in [29514 msecs]
[INFO] Building war: /Users/jjcastano/Documents/eclipse/plugin/repos/webapps/ROOT.war
[INFO] WEB-INF/web.xml already added, skipping
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ privacy-plugin ---
[INFO] Installing webapps/ROOT.war to /Users/jjcastano/.m2/repository/uc3m/privacy-plugin/1.0/privacy-plugin-1.0.war
[INFO] Installing /Users/jjcastano/Documents/eclipse/plugin/repos/pom.xml to /Users/jjcastano/.m2/repository/uc3m/privacy-plugin/1.0/privacy-plugin-1.0.pom
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 01:08 min
[INFO] Finished at: 2015-10-15T22:50:45+02:00
[INFO] Final Memory: 22M/378M
```

Si la construcción se realizó con éxito, al final de las trazas se deberá mostrar un texto que ponga "BUILD SUCCESS". En ese caso, se habrá generado en la carpeta raíz "webapps" el fichero ROOT.war, que será un war que se podrá desplegar en cualquier tomcat fuera de Eclipse.

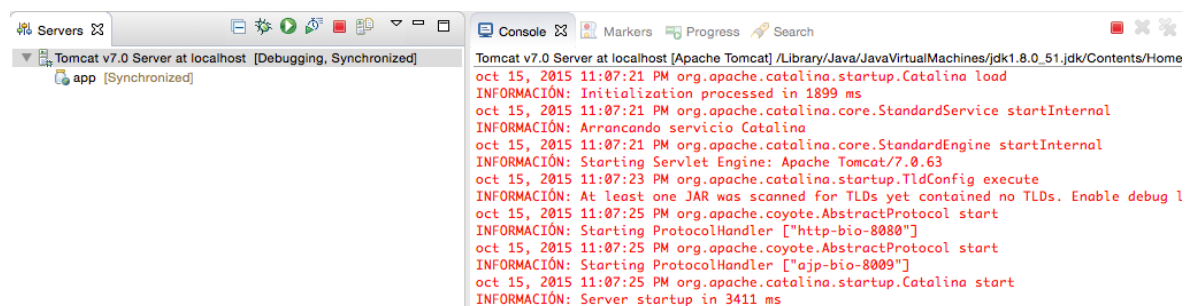
Si se quiere depurar y ejecutar la aplicación dentro de Eclipse, se deberá mostrar la vista de servidores anteriormente mencionada, seleccionar el servidor "Tomcat v7.0" y hacer click derecho. Posteriormente se deberá seleccionar "Add and Remove.."



En la ventana que se mostrará a continuación se deberá seleccionar la aplicación, pulsar el botón "Add >" para añadirlo a la zona "Configured:" y pulsar "Finish" para finalizar la configuración.



Para arrancar en modo de depuración, se deberá seleccionar el servidor y pulsar el botón Debug. Esta acción arrancará el servidor, dejando las siguientes trazas por consola:



Base de Datos

La aplicación java utiliza un conector con una base de datos MySQL. Por ello, el primer paso será instalarla.

Para ello, se debe visitar la siguiente URL y descargar la versión Community.

<http://dev.mysql.com/downloads/>



Se debe instalar tanto el servidor MySQL, como el MySQLWorkbench, que permitirá gestionar la base de datos de un forma mucho más cómoda que por consola (aunque su utilización no es imprescindible).

Una vez instaladas, se deberán crear las siguientes tablas:

browsers

Name: browsers		Schema: privacyplugin							
Column	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id	VARCHAR(37)	↕	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
creation	BIGINT(8)	↕	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
lastConnection	BIGINT(8)	↕	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

```
CREATE TABLE `activity` (  
  `id` varchar(36) NOT NULL,  
  `brwId` varchar(36) NOT NULL,  
  `timestamp` bigint(8) DEFAULT NULL,  
  `url` varchar(256) DEFAULT NULL,  
  `live` tinyint(1) NOT NULL DEFAULT '0',  
  PRIMARY KEY (`id`),  
  KEY `brwId_idx` (`brwId`),  
  CONSTRAINT `brwId` FOREIGN KEY (`brwId`) REFERENCES `browsers`  
  (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

activity

Name: activity		Schema: privacyplugin							
Column	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id	VARCHAR(36)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
brwid	VARCHAR(36)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
timestamp	BIGINT(8)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
url	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
live	TINYINT(1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
ip	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
longitude	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
latitude	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

```
CREATE TABLE `activity` (  
  `id` varchar(36) NOT NULL,  
  `brwid` varchar(36) NOT NULL,  
  `timestamp` bigint(8) DEFAULT NULL,  
  `url` varchar(256) DEFAULT NULL,  
  `live` tinyint(1) NOT NULL DEFAULT '0',  
  `ip` varchar(45) DEFAULT NULL,  
  `longitude` varchar(45) DEFAULT NULL,  
  `latitude` varchar(45) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `brwid_idx` (`brwid`),  
  CONSTRAINT `brwid` FOREIGN KEY (`brwid`) REFERENCES `browsers`  
  (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

requests

Name: requests		Schema: privacyplugin							
Column	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id	VARCHAR(36)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
activityId	VARCHAR(36)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
url	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
frameId	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
parentFrameId	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
type	VARCHAR(36)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

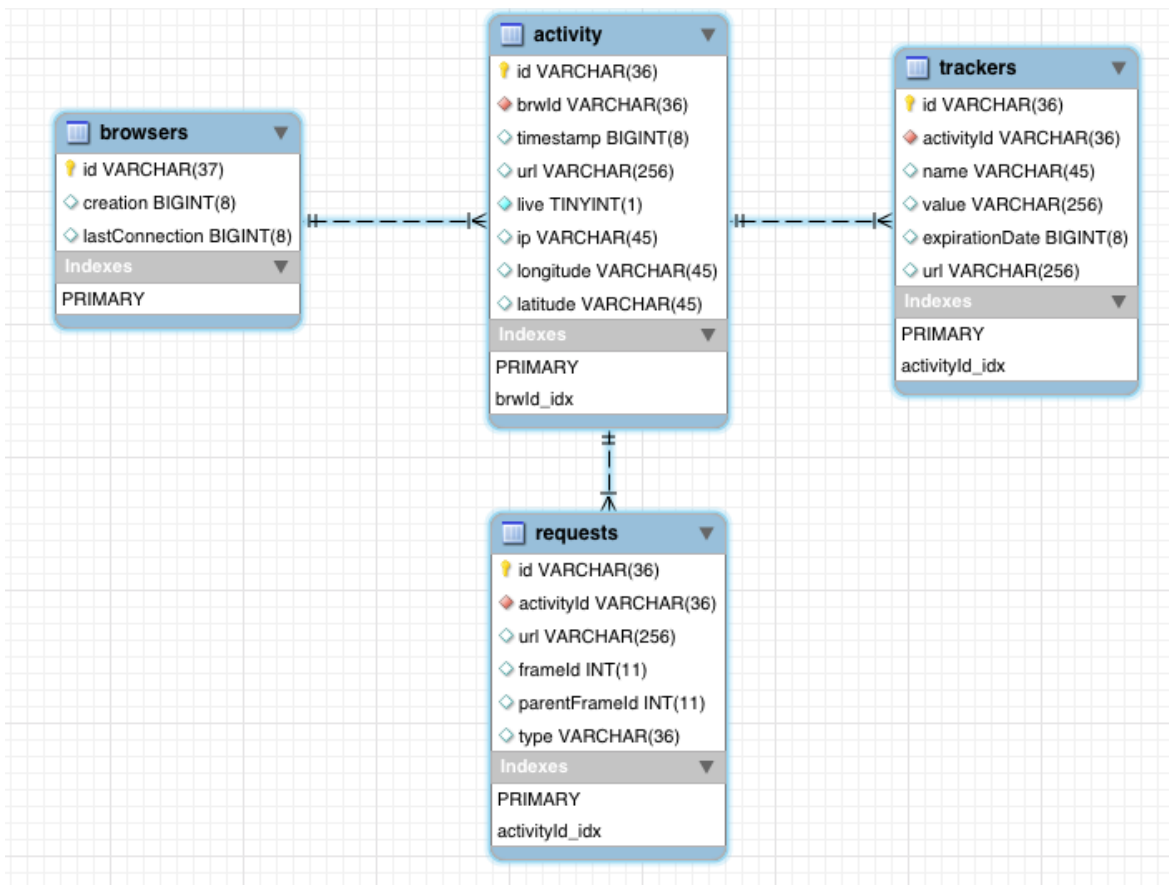
```
CREATE TABLE `requests` (  
  `id` varchar(36) NOT NULL,  
  `activityId` varchar(36) NOT NULL,  
  `url` varchar(256) DEFAULT NULL,  
  `frameId` int(11) DEFAULT NULL,  
  `parentFrameId` int(11) DEFAULT NULL,  
  `type` varchar(36) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `activityId_idx` (`activityId`),  
  CONSTRAINT `activityId` FOREIGN KEY (`activityId`) REFERENCES  
  `activity` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

trackers

Name: trackers		Schema: privacyplugin							
Column	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id	VARCHAR(36)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
activityId	VARCHAR(36)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
value	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
expirationDate	BIGINT(8)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
url	VARCHAR(256)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

```
CREATE TABLE `trackers` (
  `id` varchar(36) NOT NULL,
  `activityId` varchar(36) NOT NULL,
  `name` varchar(45) DEFAULT NULL,
  `value` varchar(256) DEFAULT NULL,
  `expirationDate` bigint(8) DEFAULT NULL,
  `url` varchar(256) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `activityId_idx` (`activityId`),
  CONSTRAINT `id` FOREIGN KEY (`activityId`) REFERENCES `activity`
(`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

El modelo final es el siguiente:



ANEXO C - Manual de instalación del interfaz web

La interfaz web del proyecto está desarrollada en HTML5, JS (utilizando dojo como framework y jQuery como conjunto de utilidades) y CSS3 (utilizando less como lenguaje intermedio y bootstrap como utilidad).

La entrega del software que se anexa junto a esta práctica no incluye las dependencias de terceros, por lo que será necesario descargarlas para su correcto funcionamiento.

Para ello, será necesario tener instalado bower y nodejs.

nodejs

Nodejs es un motor de ejecución js construido sobre el motor V8 de Chrome que permite ejecutar módulos js en una máquina y se utiliza para automatizar tareas, desarrollar el back end de aplicaciones web..

Se puede descargar desde el siguiente enlace:

<http://nodejs.org>

Una vez instalado se deberán descargar las dependencias necesarias. Para ello, se deberá abrir un terminal de comandos y situarse en la carpeta *src/main/webapp/libs*. Una vez allí se deberá ejecutar el siguiente comando:

```
npm install
```

Al hacer esto, dentro de esta carpeta se creará una nueva carpeta llamada *node_modules* con todas las dependencias del proyecto en su interior.

bower

Bower es un gestor de dependencias de software de front que permite descargar cualquier librería que tenga indexada.

Se puede consultar como descargar desde el siguiente enlace:

<http://www.bower.io>

O, si se ha instalado previamente nodejs, ejecutando:

```
npm install -g bower
```

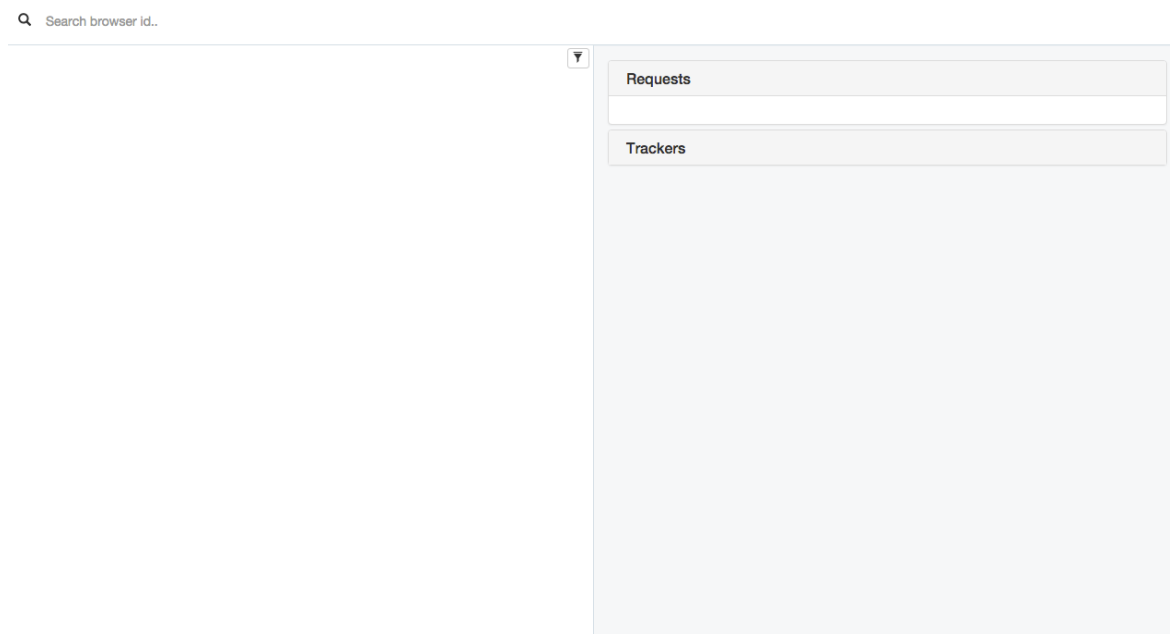
Una vez instalado se deberán descargar las dependencias necesarias. Para ello, se deberá abrir un terminal de comandos y situarse en la carpeta *src/main/webapp/libs*. Una vez allí se deberá ejecutar el siguiente comando:

```
bower install
```

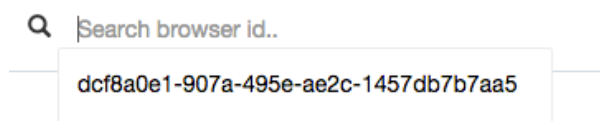
Al hacer esto, dentro de esta carpeta se creará una nueva carpeta llamada *bower_components* con todas las dependencias del proyecto en su interior.

ANEXO D - Manual de usuario del interfaz web

La aplicación muestra un interfaz web muy sencillo con una única pantalla que permite seleccionar un browser y un rango de fechas.



Para poder ver todos los browsers y seleccionar uno, dicha acción se realizará pulsando el combo de la esquina superior izquierda y teclear "Ctrl + SPACE".



Seleccionando uno de los browsers, en la zona izquierda se cargarán todas las páginas que dicho browser ha visitado, siendo estas todas las que se visitaron antes de instalar el plugin (recolectadas desde el histórico) y todas las posteriores, con monitorización activa del plugin.

Las columnas mostradas serán la URL de visita, el timestamp de cuando se visitó y un booleano indicando si la visita fue durante una monitorización activa o no.

Si se pulsa la cabecera de la tabla, ésta se ordenará alfabéticamente por dicha columna.

En el pie de la misma, aparecerá el numero de registros y la paginación que se puede realizar en los datos.

Q dcf8a0e1-907a-495e-ae2c-1457db7b7aa5

URL	Timestamp	Live
https://www.youtube.com/?hl=es&gl=ES	1443305108947	false
https://github.com/theintern/intern/tree/geezer	1439221086150	false
https://www.google.es/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=...	1443523238165	false
http://tympanus.net/codrops/2012/10/23/basic-ready-to-use-css-styles/	1438373048732	false
http://tympanus.net/Development/PageTransitions/	1439197520420	false
http://descargasmix.com/peliculas/hd/m720p/page/21/	1443308500865	false
https://www.google.es/maps/@40.4562239,-3.8146545,3a,75y,215.14h,76.48t/data=!3...	1441814342180	false
http://tympanus.net/Tutorials/StickyTableHeaders/index3.html	1438365158318	false
http://tympanus.net/Development/DialogEffects/cathy.html	1443476318080	false
http://tympanus.net/Development/SelectInspiration/index8.html	1443477044599	false
https://www.google.es/search?q=Sus+preferencias+de+seguridad+permiten+la+instala...	1442274400627	false
http://tympanus.net/codrops/2013/02/26/elastic-content-slider/	1438371504896	false
https://www.google.es/search?q=dojo+domgeom&dq=dojo+domgeom&aqs=chrome..6...	1438690123179	false
https://www.google.es/ur?sa=&rc=j&q=&src=s&source=web&cd=1&ved=0CCIQFjAA...	1439302582321	false
https://www.google.es/maps/@40.4600652,-3.812761,19.49z	1441814555296	false
http://tympanus.net/codrops/2013/05/21/natural-language-form-with-custom-input-ele...	1438371439938	false
http://tympanus.net/codrops/2012/10/22/slicebox-revised/	1438373053308	false
http://descargasmix.com/como-acabar-sin-tu-jefe-2-bdrip-m720-castellano-comedia-2...	1443308058861	false
http://tympanus.net/codrops/2013/02/15/fresh-and-creative-web-design-techniques/	1438371506759	false
http://tympanus.net/Tutorials/AppShowcase/index3.html	1438372570852	false
https://www.jstree.com/	1438604751116	false
https://www.google.es/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=...	1437680774152	false
http://java2s.com/Code/Jar/CatalogJar.htm	1439284376349	false
https://www.google.es/search?q=dojox.html&dq=dojox.html&aqs=chrome..6957j0l5.39...	1438694673205	false
http://stackoverflow.com/questions/25604969/setting-a-system-variable-within-a-mave...	1439304945583	false

901 - 1000 de 3557 resultados

Para realizar un filtrado en ellos, se podrá realizar pulsando en el botón superior izquierdo de dicha tabla

URL	Timestamp	Live
https://www.youtube.com/?hl=es&gl=ES	1443305108947	false
https://github.com/theintern/intern/tree/geezer	1439221086150	false
https://www.google.es/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=...	1443523238165	false
http://tympanus.net/codrops/2012/10/23/basic-ready-to-use-css-styles/	1438373048732	false
http://tympanus.net/Development/PageTransitions/	1439197520420	false
http://descargasmix.com/peliculas/hd/m720p/page/21/	1443308500865	false
https://www.google.es/maps/@40.4562239,-3.8146545,3a,75y,215.14h,76.48t/data=!3...	1441814342180	false
http://tympanus.net/Tutorials/StickyTableHeaders/index3.html	1438365158318	false
http://tympanus.net/Development/DialogEffects/cathy.html	1443476318080	false
http://tympanus.net/Development/SelectInspiration/index8.html	1443477044599	false
https://www.google.es/search?q=Sus+preferencias+de+seguridad+permiten+la+instala...	1442274400627	false
http://tympanus.net/codrops/2013/02/26/elastic-content-slider/	1438371504896	false
https://www.google.es/search?q=dojo+domgeom&dq=dojo+domgeom&aqs=chrome..6...	1438690123179	false

Filtro

Fecha de inicio:
dd/mm/aaaa

Fecha de fin:
dd/mm/aaaa

☐ Live

Aplicar

Se deberá introducir una fecha de inicio, otra de fin y seleccionar o no el check "Live."

Si se selecciona, sólo se mostrarán los registros con monitorización activa. Si no está seleccionado, se mostrarán todos. Si se selecciona una actividad sin monitorización activa, el detalle que se muestre en la parte derecha se mostrará vacía (pues el plugin no pudo recolectar información)

Si se selecciona una actividad con monitorización activa, la zona izquierda se cargará con el detalle de la información recolectada durante aquella visita.

dcf8a0e1-907a-495e-ae2c-1457db7b7aa5

URL	Timestamp	Live
http://www.forocoches.com/	1444868705160	true
https://www.google.es/_/chrome/newtab?espv=2&ie=UTF-8	1444868703542	true

Requests

URL	FrameId	ParentF...	Type
https://c.betrad.com/a/n/366/9595.js	28	19	script
http://www.forocoches.com/js/add2home_ipad.js	0	-1	script
https://googleads.g.doubleclick.net/pagead/ads?client=ca-pub-2852522843813544&output=html&h=90&slotname=3141121293&ad...	19	17	sub_frame
https://cm.g.doubleclick.net/pixel?google_nid=exp&google_push=AHNF13LzmH7s5RRtQmonSI_DvelC...1y-X3Pu&google_ula=Z786954&google_hm=18072662575594009220	25	14	image
https://bid.g.doubleclick.net/xbbe/creative/ad?d=APEucNVApz7Gb6Ljceep17bWXYVgZLg5WLMMLJNNPKIndpzi...Q8BWYnYUk-edN5tO_D3WEGvURPy2LEMxe7Qib_SVfvt_-3580n09YXFAS3Z0EG...w1V9g-MacVY	14	0	script
http://www.forocoches.com/js/add2home.css	0	-1	stylesheet
https://bs.serving-sys.com/BurstingPipe/adServer.bs?cn=rs&c=28&pl=14764257&Puid=0&w=300&h=600&nou=http://ad...	14	0	script

1 - 100 de 125 resultados

Trackers

La información se divide en dos zonas, una de Requests, donde estarán los recursos que se descargaron y las url's que se cargaron en iframes internos (relevantes para el estudio, pues en ellos se cargarán las campañas publicitarias)

Las columnas mostradas serán la URL, el identificador del frame que realizó la petición, el identificador del frame padre y el tipo de recurso.

La zona inferior es la de trackers:

dcf8a0e1-907a-495e-ae2c-1457db7b7aa5

URL	Timestamp	Live
http://www.forocoches.com/	1444868705160	true
https://www.google.es/_/chrome/newtab?espv=2&ie=UTF-8	1444868703542	true

Requests

Trackers

URL	Name	Value
www.forocoches.com/	/	0 sc1
.forocoches.com/	__utma	119157022.21...
.forocoches.com/	__utmc	119157022
.wtp101.com/	tuuid	9ec39401-a4ff-4167-8c4a-955259237cac
.antevenio.com/	vs	58395=6724906
.serving-sys.com/	u2	357ad842-85fa-4346-a4ea-1de9369b58be...
.tribalfusion.com/	ANON_ID	aSnZdWKN3ld...
.forocoches.com/	__utma	119157022.21...
.forocoches.com/	__utmz	119157022.14...(direct) utmcm=...

1 - 26 de 26 resultados

En ella se mostrará la información referente a las cookies que se pusieron en dicha página, mostrando la información de la URL desde donde se puso la cookie, su nombre y su valor.